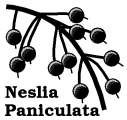
Using generative probabilistic models for multimedia retrieval

USING GENERATIVE PROBABILISTIC MODELS FOR MULTIMEDIA RETRIEVAL

Thijs Westerveld



Taaluitgeverij Neslia Paniculata

Samenstelling van de promotiecommissie:

Prof. dr. F.M.G. de Jong, Universiteit Twente (promotor)
Dr. A.P. de Vries, CWI (assistent-promotor)
Prof. dr. W.H.M. Zijm, Universiteit Twente (voorzitter/secretaris)
A. Hauptmann PhD, Carnegie Mellon University
Prof. dr. M.L. Kersten, Universiteit van Amsterdam
Dr. P.A.M. Kommers, Universiteit Twente
Prof. dr. ir. A. Nijholt, Universiteit Twente
Prof. dr. A. Smeaton, Dublin City University
Prof. dr. A.W.M. Smeulders, Universiteit van Amsterdam

Taaluitgeverij Neslia Paniculata
Uitgeverij voor Lezers en Schrijvers van Talige Boeken
Nieuwe Schoolweg 28, 7514 CG Enschede, The Netherlands

CTIT Ph.D.-series No. 04-67
Centre for Telematics and Information Technology
PO Box 217, 7500 AE Enschede, The Netherlands

SIKS Dissertation Series No. 2004-19
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

CIP GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Westerveld, Thijs

Using generative probabilistic models for multimedia retrieval/
T. Westerveld
Enschede: Neslia Paniculata
CTIT Ph.D.-thesis series, ISSN 1381-3617; No. 04-67
Thesis Enschede – with ref. – with summary
ISBN 90-75296-13-4
Subject headings: information retrieval

Printed by Print Partners Ipskamp, Enschede.
Cover photo by Jelger Bakker.

This thesis is available online at `http://purl.org/utwente/41716`.

USING GENERATIVE PROBABILISTIC MODELS FOR MULTIMEDIA RETRIEVAL

P R O E F S C H R I F T

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 25 november 2004 om 13.15 uur

door

Thijs Henk-Willem Westerveld
geboren op 23 september 1974
te Gendringen

Dit proefschrift is goedgekeurd door de promotor,
prof. dr. F.M.G. de Jong,
en door de assistent-promotor,
dr. A.P. de Vries.

Contents

Acknowledgements

My interest for research was raised in the last few years of my studies of computer science at the University of Twente. In those years, I most enjoyed the courses related to language technology taught by Anton Nijholt and Franciska de Jong's group, then called Parlevink. I am grateful that Franciska offered me to work on the twenty-one and pop-eye projects after I finished my master's thesis and will never regret having accepted that position. In the first years of my research career I was lucky enough not to have to find a PhD subject immediately. Instead, I could develop my research skills and interests by working on a number of projects funded by Dutch and European government. Within these projects I got to travel around Europe for numerous meetings, that were sometimes boring, but often provided inspiring technical discussion. Also in this period, I had the privilege to attend workshops, conferences and summer schools.

Having started my research career in text retrieval, my interest and the projects I worked on gradually shifted to imagery: at first I concentrated on retrieving visual material via associated text, but after a while I started looking into the disclosure of images based on their visual content. To learn more about image processing, I followed two of Ferdi van der Heijden's courses: 'digital image processing' and 'introduction to pattern classification'. Together with Djoerd Hiemstra's success with a probabilistic, language modelling approach to text retrieval, these two courses raised my enthusiasm for decision theory, pattern classification and probabilistic models. Soon, I started implementing probabilistic models for images. I thank Franciska for allowing me to explore this that was not only new to me but to the whole group. Now, at CWI I am in a similar position. I believe I am the first person to work two years in INS1 without using MonetDB. Although at times I wished everything were in a database, I think the continued use of the old matlab scripts was the best choice for a smooth path towards a PhD

dissertation.

I am indebted to Arjen de Vries for numerous lively discussions on probabilistic modelling of multimedia retrieval. Also the small reading club we had with Jan-Mark Geusenbroek and Jan van Gemert when I just came to Amsterdam taught me a lot. Reading those papers on complicated probabilistic models together was fun and allowed me to understand the models better than I could have on my own. I also enjoyed the lively discussions with colleagues both at home and abroad.

Both at the University of Twente and at CWI, I could combine dissertation research and project work. The work presented in this thesis has been partially funded by the NWO/SION project 'a picture tells' and the SENTER project 'waterland'.

During the last months of finishing my PhD trajectory and writing this book, my promotor Franciska de Jong and assistant-promotor Arjen de Vries have been of invaluable support. Without their encouragement and their devotion to reviewing finished and half-finished chapters, this book would not have been written. I am honoured that Alan Smeaton, Alex Hauptmann, Arnold Smeulders, Martin Kersten, Anton Nijholt and Piet Kommers agreed to participate in the graduation committee.

Christian Bakker and Dirk van Essen deserve credit for providing me with different variants of the jigsaw metaphor that allowed me to elegantly explain the nature of the generative models. I thank Peter Grünwald for reviewing early versions of Chapters 3 and 5 and to Marlies Koffeman and Jessica Nash for proof reading portions of the book. Of course all remaining errors are mine.

I thank my family, friends and colleagues for their interest and support, and for distracting me from my daily business. Those dinners, coffees, drinks, parties, films and concerts were great. My parents deserve special mentioning, not only for playing 'kleur bekennen' when I was young, the game that provided the dice on the cover of this book, but moreover for their encouragement and belief in me. Finally, I would like to thank Jelger for always being there. His faith and confidence have motivated me greatly.

Thijs Westerveld

Amsterdam, October 2004

1

Introduction

Information retrieval is the field concerned with the structure, analysis, organisation, storage, searching and retrieval of information (Salton, 1968). Although this early definition of the field is very broad, traditionally information retrieval has focused on the retrieval of textual documents. This thesis focuses on ad hoc retrieval from heterogeneous multimedia archives. At least three concepts need further introduction: *ad hoc retrieval*, *multimedia archives* and *heterogeneous archives*.

Ad hoc retrieval is the task of searching a static collection for relevant documents given an information need. Relevant documents are those documents that satisfy the information need. In ad hoc retrieval, no prior knowledge is available on the relevance of documents. The system can only use the description of the information need, which is often called *query* or *topic*. Typical examples of ad hoc search systems are the search engines on the web (e.g., Google[1], AltaVista[2]).

Multimedia archives are collections of multimedia documents. Each of the documents in the collection can contain multiple media and be a mixture of text, images, video and audio. However, the term *multimedia* is often also used to refer to a single medium, provided that it is not text. We adopt this convention and use the term multimedia document to refer to any document containing at least an image, a piece of video material, or an audio fragment.

Heterogeneous archives consist of documents from a broad domain. A huge variability in topics can be expected, and apart from the document format (e.g., ascii texts, html pages, jpeg images) often nothing is known about the structure of the documents. For example, within collections of ascii texts, or jpeg images, a wide range of document sizes may be found, and a document in a collection of html pages may contain anything from a

[1] `http://www.google.com`
[2] `http://www.altavista.com`

few lines of text, to a single image, to a full multimedia document. This contrasts with *homogeneous archives*, collections of documents from a narrow domain, where many properties of the documents can be predicted and little variation exists between documents. Examples of homogeneous multimedia archives are medical image databases, collections of fingerprints and archives of frontal views of human faces. The set of images on the world wide web is an extreme example of a heterogeneous multimedia archive. Other heterogeneous document collections, may be more predictable in structure, but still cover a broad range of topics. An example of this is a news archive.

Ad hoc retrieval from heterogeneous archives is a difficult task. No prior knowledge about the topics covered in the collection is available and nothing is known about the information need or the user before the retrieval session starts. One way of dealing with this lack of knowledge is to restrict the search possibilities to a fixed number of concepts that are known to be identifiable. Another way is to require more effort from the user. For example, a user could give additional information about his need by presenting a number of relevant documents to the system, or by giving feedback on the system's output in an iterative process. In this work, we look for generic solutions that are applicable in a broad domain and require little or no user effort. For example, the solutions should not require the manual labelling of large amounts of training data. This makes the solutions flexible, fit for a broad range of information needs and easily transferable to new domains.

1.1 The basic information retrieval framework

The main task of an information retrieval system is to identify relevant documents, which satisfy a user's information need that is expressed by a query. Since direct access to document or query content at a semantic level is impossible, information retrieval systems need to work with content representations. The commonly adopted framework is visualised in Figure 1.1 (e.g., Fuhr, 1992; Croft, 1993). The user's information need is expressed by a query, while possibly relevant information appears in the form of documents. The retrieval system transforms query and documents to internal representations and based on these decides whether a document is relevant to a given query or not. These decisions typically result in a *retrieval status value* (RSV) for each pair of query and document representations. These RSVs can be viewed as the scores of the documents for the query, or the confidence the system has in the relevance of the document. The documents with the highest RSVs are returned to the user.

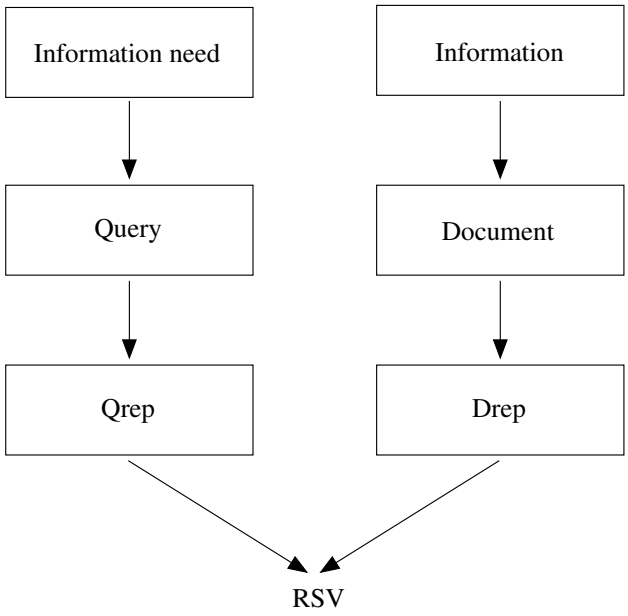


Figure 1.1: Information retrieval framework

1.1.1 Uncertainty in representations

The main problems in information retrieval are related to the different types of uncertainty in the representations of information need and document content. First, a query may be an incomplete and imprecise description of the information need. Second, different documents may use different syntactic elements to describe the same semantic concepts. In text retrieval, this second problem is sometimes referred to as the paraphrase problem (Oard and Dorr, 1996). These sources of uncertainty do not belong to the retrieval system proper, but they complicate the task. The system only has direct access to the query and the document appearance, and never directly to information need or document content. Nevertheless, this level of indirectness and the uncertainty involved must be taken into account. The aim of an information retrieval system is to find documents that are relevant at the content level, i.e., the document content should be relevant to the information need. Therefore, the second transformation, from query and document appearance to internal representations, should capture the information on the content level.

In this thesis, collections consist of multimedia documents. The documents appear either as images or as video fragments, possibly accompanied by text. The assumption throughout this thesis is that for the document content, i.e., for the information gathered from the document, the visual as-

Figure 1.2: Keyframe of video fragment from CNN news broadcast

pect matters. For example, the information a user gathers from the video fragment from which Figure 1.2 shows a frame, can be 'anchor person', or 'studio setting' or 'woman in pink suit', or 'Lynn Vaughn' but not 'Helicopter crash' even though that is the subject of the news item.

For representing the visual aspects of multimedia documents, two approaches are common. The first approach relies on external (textual) descriptions of the visual information, the second builds descriptions from the visual document appearance directly.[3] Typically, the format in which a user can formulate a query depends on the choice of internal document representation. When a textual representation is used, text-based querying is offered; when document representations are based on the visual content, queries should be represented visually (see also Section 2.3). Regardless of the approach taken, the internal representations of query and document are bound to introduce more uncertainty, because the information the system gathers from a document will differ from the information the user gathers from it.

The problems with uncertainty of representations can be summarised as follows. The retrieval system has no direct knowledge of either the information contained in the document or the user's information need, let alone of the relevance relation between the two.

1.1.2 Probabilistic modelling to deal with uncertainty

A natural way of dealing with representation problems that has become popular in the field of text retrieval is known as the language modelling approach

[3] A similar distinction can be made in text retrieval. Traditional library approaches use external descriptions of the document content (keywords, concepts), while modern full-text approaches build representations directly from the text in the documents.

to information retrieval (Ponte and Croft, 1998; Hiemstra, 1998; Miller et al., 1999). In this approach, each document is represented as a generative probabilistic model, i.e., a distribution over terms that describes the document's 'language'. Generative probabilistic models describe how likely it is for each term to occur in the document; typically care is taken to assign non-zero probabilities to all terms in the vocabulary. Of course, using probabilistic models does not solve the representation problems, but at least the models capture some aspect of the uncertainty in the representation: each term could have been present, but some are more likely to occur than others. The models have proved successful on a variety of information retrieval tasks including ad hoc retrieval.

Independent of the language modelling approach to information retrieval, generative models for content-based image retrieval have been proposed (e.g., Vasconcelos, 2000; Fergus et al., 2003; Greenspan et al., 2001; Luo et al., 2003). The generative probabilistic image models define a probability distribution over visual features, like the generative language models define a distribution over terms. The assumption in both variants is that the query is an observation generated from a document model in the collection. The RSV for a document is then estimated by the probability that the document model generates the query.

Not only are generative probabilistic models an elegant way of dealing with the uncertainty in representations, probabilities also have the nice quality that they are well-defined, independent of media or domain. This is particularly useful in the field of multimedia retrieval, since probabilities obtained from representations in different media can be compared. A probabilistic framework allows for the seamless integration of probabilities obtained from for example, text, image, audio and video.

In this thesis generative models are used for representing both textual and visual parts of the multimedia documents. They will be applied to ad hoc multimedia retrieval separately and in combination. Parallels between the textual and visual models will become clear and techniques from the textual approach will be applied to the visual models.

1.2 Evaluation

As the first word in the title of this thesis suggests, the generative probabilistic models will be *used* in an experimental setting. They will be tested and evaluated on the task of ad hoc retrieval from multimedia collections.

As exemplified by the popularity of TREC[4] and the large proportion of

[4]Text REtrieval Conference, a workshop series for large scale evaluation of information

experimental papers in SIGIR[5], experimentation is widely acknowledged as one of the driving forces behind the advancement of information retrieval. In his keynote speech at SIGIR 2003, Croft (2003) memorised the importance of experimental results in the field of information retrieval as follows.

> The information retrieval community will not accept a model, however nice or mathematically correct, unless it is backed up by experimental results. At least an honest attempt to measure is needed.

In content-based image retrieval however, evaluation does not have a long tradition. Evaluation of content-based image retrieval has focused on testing techniques, rather than their usefulness in a retrieval setting. Many content-based image retrieval techniques have been claimed successful, but these claims are often based on observations in limited domains (e.g, Flickner et al., 1997; Smith and Chang, 1997; Duygulu et al., 2002; Jeon et al., 2003). Image retrieval systems are typically evaluated on datasets with clear distinctions between subsets of homogeneous images. In such a setting, where the collection consists of clearly defined distinct sets of for example *sunsets*, *zebras* and *aeroplanes*, it is relatively easy to find other examples of say *sunsets*. Experiments in such settings can be useful, but one has to be careful not to draw too general conclusions from them. Only recently the field has become concerned with evaluation methods and collections (Smeaton et al., 2003b; Smeaton and Over, 2003; Müller, 2002; Müller et al., 2001; Gunther and Beretta, 2000). Still, even to date, many papers do not evaluate their results beyond showing a few, well-chosen examples.

1.3 Research objectives

The research presented in this thesis is motivated by three issues.

First, we investigate how generative probabilistic models can be used for multimedia retrieval. We take existing generative probabilistic models from the domains of text retrieval and image retrieval and describe how they can be used separately and in combination. Applying formal models to multimedia retrieval has been identified as one of the main challenges in multimedia retrieval (Allan et al., 2003). This is exactly the challenge faced in this thesis.

---

retrieval technology

[5]Annual international ACM SIGIR conference on research and development in information retrieval

Second, we regard content-based retrieval, or multimedia retrieval in general, as a subset of information retrieval. There may exist many differences between multimedia documents and textual documents, but in essence multimedia retrieval – like text retrieval – boils down to finding information. Therefore, it is important to acknowledge the rich history of text retrieval and learn from it. In this thesis, we identify parallels between the generative model approaches to text retrieval and to visual retrieval. This allows us to draw from the ideas that are known to work well in the language modelling approach to text retrieval and apply them to multimedia retrieval.

Third, we acknowledge the importance of experimental results and evaluate the proposed methods using sizable collections. Since the generative probabilistic models are intended for ad hoc search in heterogeneous collections, we experiment with large collections from a broad domain. We adopt the evaluation methodology and metrics that are well-established in the field of text retrieval and use them to evaluate multimedia retrieval. We also reflect on the usefulness of the common evaluation methodology in the multimedia domain.

Summarising, this thesis addresses the following issues.

- How can generative probabilistic models be applied to multimedia retrieval?
- Can we identify and leverage parallels between the use of generative models for multimedia retrieval and similar approaches to text retrieval?
- How do the techniques based on generative models perform on the task of ad hoc retrieval from a generic collection?

1.4 Thesis overview

This thesis is organised as follows.

Chapter 2 discusses related work in the relevant fields. It gives a brief overview of text retrieval models and the use of text retrieval techniques for multimedia retrieval. It then discusses content-based retrieval techniques, focusing on probabilistic methods and methods that are inspired by text retrieval. The chapter concludes with a short introduction to the evaluation methodology used throughout this thesis.

Chapter 3 provides an introduction to generative probabilistic models that are at the basis of the present work. Step by step, it explains the nature of the models, how they can be used in information retrieval and how

the model parameters can be estimated. The chapter closes with a short discussion on the relation between generative models and relevance.

Chapter 4 presents experimental results. The generative models are evaluated on a number of commonly used test collections. We set a baseline by using the generative probabilistic models with minimal user input. We then investigate whether it is possible to improve upon this baseline by manually selecting 'good' query examples, by selecting regions within query examples, or by multi-modal querying, i.e., combining textual and visual data in a single query.

Chapter 5 covers variants and extensions of the generative probabilistic models. The chapter discusses differences between so-called *query generation* and *document generation* variants, and investigates how to model only the distinguishing aspects of documents rather than the whole document. The chapter also describes Bayesian extensions to the proposed models and relates the models to other models in the literature. All variants presented are shown to stem from a single probabilistic framework, and to differ only in the way probabilities are estimated.

Chapter 6 reflects on the evaluation methodology that has been applied to multimedia retrieval in this thesis and elsewhere. The chapter starts with a brief history of information retrieval evaluation and a description of the Cranfield tradition, the commonly used paradigm in text retrieval evaluation. The chapter then discusses how this paradigm can be extended to evaluate retrieval from multimedia collections.

Finally, Chapter 7 summarises our main findings, discusses our contributions to the field and identifies possibilities for future research.

2

Related work

The background for the rest of this thesis is provided here. Section 2.1 introduces information retrieval. Section 2.2 and 2.3 discuss text-based and content-based access to multimedia material respectively. Section 2.4 discusses approaches that combine textual and visual information. Section 2.5 introduces the basic mechanisms for evaluating information retrieval. Finally, Section 2.6 discusses how the work in this thesis relates to previous work. A thorough introduction to the field of probability theory can be found in many places (e.g, Jaynes, 2003; Sivia, 1996).

2.1 Information retrieval models

The task of an information retrieval system is to identify relevant documents given a user's information need. As we have seen, information retrieval systems cannot access information need and document content directly, but have to rely on representations of them. In essence, information retrieval is about representing documents and queries, and about comparing the representations to determine if a relevance relation exists (cf. Figure 1.1). Hence, a retrieval model needs to specify a query representation method, a document representation method and a function to compute the retrieval status value based on the two representations. The following subsections briefly describe the most common models. An extensive review of retrieval models is beyond the scope of this thesis and can be found elsewhere. For example, Sparck Jones and Willett (1997b) survey information retrieval models that were proposed between the mid-seventies and mid-nineties. The survey is followed by a selection of the original papers. Detailed descriptions of many retrieval models, including the more recent language modelling approach to information retrieval, can be found in Kraaij's PhD thesis (Kraaij, 2004).

2.1.1 Boolean model

In a Boolean retrieval system a user connects search terms using the logical operators AND, OR and NOT. The system then returns the documents that fully satisfy the logical constraints of the query. The RSV for a document in a Boolean retrieval system is either 1, if the constraints of the query are met by the document, or 0, otherwise. For example, the query *(image* OR *video)* AND *retrieval* will return documents that contain the term *retrieval* and either the term *image* or the term *video*. The Boolean model assumes that all documents that meet the logical constraints of the query are equally relevant to the information need. This means Boolean system return unordered sets of documents and the user does not get a handle on where to start examining the documents. In principle, the whole returned set has to be consulted. As pointed out by Cooper (1988) this can be problematic, since these sets can be very large, especially for short and simple requests in large collections. Longer requests are harder to create for novice users and often return no documents at all.

2.1.2 Ranked retrieval models

An alternative to Boolean retrieval that overcomes most of these limitations is ranked retrieval. In a ranked retrieval setting, systems do not return an unordered *set* of documents, but an ordered *list*. Intuitively, the best strategy is to put the documents that are most likely to be relevant at the highest ranks. Robertson (1977) has shown that indeed – if documents are treated independently – the optimal ordering of returned documents is by decreasing probability of relevance. This is known as the Probability Ranking Principle. The estimates of the probability of relevance can be based on probabilistic principles, but also on other principles (e.g., the distance in a vector space). The models discussed in the remainder of this section are all ranked retrieval models.

Vector space model The vector space model (Salton et al., 1975) represents queries and documents as vectors in a high dimensional space. Each of the dimensions of the space corresponds to a term in the vocabulary. The vocabulary is the list of words that are used for representing documents and queries. Often this is simply the union of all words in the collection (possible after linguistic processing). In the vector space model, queries and documents are represented as vectors of term weights. In the simplest case, these weights could be binary and represent if a vocabulary term is present or absent in a document, but typically *tf.idf* weighting is used. In such a

weighting scheme, the weight of a term is proportional to the frequency of the term in a document (term frequency, tf) and inversely proportional to the number of documents the term occurs in (inverse document frequency, idf). The rationale for this is that terms that are frequent in a document are important indicators of the document's content, while terms that are frequent throughout the collection are not.

The matching function in the vector space model is a function of the weights vectors for query and document. Often, the cosine of the angle between the vectors is used. The cosine measure ignores document length, thus allowing for ranking of documents of varying length.

Latent semantic indexing The idea behind latent semantic indexing (Deerwester et al., 1990) is that terms that often occur in the same documents are semantically related. Similarly, documents that share many terms are likely to be on the same subject. Latent semantic indexing starts from the vector space model representation: queries and documents are represented as weight vectors in the high dimensional space spanned by the vocabulary terms. The dimensionality of the space is then reduced by computing the most meaningful linear combinations of terms and documents using singular value decomposition. In the resulting space, related terms, and similar documents will be close to each other. Matching of query and documents is performed in the lower-dimensional space. A probabilistic variant of latent semantic indexing was proposed by Hofmann (1999). We will return to this probabilistic variant in Chapter 5 of this book, where its relation to the models applied in this thesis is explained.

Binary independence retrieval model The binary independence retrieval model (Robertson and Sparck Jones, 1976) aims at directly estimating the odds of relevance given a query and document representation. The assumption is that term frequencies in relevant documents differ from those in non-relevant documents. Using the binary independence assumption (terms occur independently in documents), the score for a document can be computed as the product of the individual term scores. Cooper (1991) showed that it suffices to assume linked dependence rather than independence. The linked dependence assumption states that the likelihood ratio of a set of terms given relevance and non-relevance can be computed as a product of the individual term ratios. To compute these term ratios, the likelihoods of occurring in relevant and non-relevant documents have to be estimated for each term in the vocabulary. Once relevance information is available, these estimates can easily be obtained. Without relevance information however, the model

performs poorly.

Language models for retrieval The language modelling approach to information retrieval (Ponte and Croft, 1998; Hiemstra, 1998; Miller et al., 1999) represents each document as a generative statistical models of terms. Each document model defines a probability distribution over the terms in the vocabulary. Queries are represented as terms and are assumed to be observations from a document model. The RSV for a document is calculated by estimating the probability that a document model generates the query terms. As language models are used for modelling textual data throughout this thesis, they are discussed in detail in Chapter 3.

2.2 Text retrieval for multimedia collections

Multimedia retrieval is a specialisation of information retrieval. Multimedia documents may be different from textual documents, but the underlying task is the same: identifying relevant documents to satisfy some information need. To achieve this goal, representations of the query and document are needed. Since multimedia documents are different from text documents, different representations are needed. These representations can be built from an external textual description of the content, as discussed below, or from the document's visual content directly (see Section 2.3).

One way to disclose multimedia collections is to take the traditional library approach and manually construct representations of the multimedia documents by assigning descriptive terms to each document. This manual annotation approach is still used in many multimedia archives.(e.g., Corbis[1], Getty images[2], ANP beeld[3], Beeld en Geluid[4]). But, manual annotation is expensive and it requires a lot of training to do consistently. An alternative that allows for automatic disclosure, is to exploit collateral text. Many multimedia objects come with textual data. For example news paper photographs have captions, web images have surrounding text and film or video material often has subtitles. These related texts are exploited by most web search engines nowadays, even by the ones specifically targeted at multimedia material (e.g., Google's image search[5], or AltaVista's[6] image, video and

[1] http://www.corbis.com
[2] http://www.gettyimages.com
[3] http://www.anp.nl/beeld
[4] http://www.beeldengeluid.nl
[5] http://www.google.com/images
[6] http://www.altavista.com/

audio search).

For video and audio search, web search engines rarely go beyond retrieving full video or full audio documents. However, long documents may contain many different aspects, thus it is useful to get a pointer to the exact position in the media where the relevant information is. To facilitate this, time-coded representations of the data are needed. Such representations need to provide a minute-by-minute (or second-by-second) description of the media content. Again, manual annotation can be used to get such a detailed description, and in fact media archives do this. But, a less laborious approach is to apply speech recognition to the audio signal and use traditional text retrieval techniques on the resulting time-coded speech transcripts (Witbrock and Hauptmann, 1998; Garofolo et al., 2000; Abberley et al., 1998; De Jong et al., 2000; Ordelman, 2003).

Using manual annotations or collateral text for the disclosure of multimedia collections has its limitations. First of all, textual information is not always available. While produced data, like newspaper images and broadcast video often come with linguistic elements (e.g., captions, speech), raw data, like unproduced footage and personal digital photo archives, usually lack this information. But even when textual information is available it can never describe everything that is present in the multimedia document. Thus, disclosure is limited to what happens to be described. An alternative approach, that overcomes the limitations of external textual descriptions, is to work with the visual content directly and to deduce information from the pixel values. This approach is discussed in the next section.

2.3 Content-based visual retrieval

The field that represents documents based on visual characteristics rather than on external descriptions is referred to as *content-based image retrieval*, or sometimes shortly *content-based retrieval*. Since these terms are too narrow and too broad respectively, we will use the term *content-based visual retrieval* to refer to techniques that build document and query representations (for image or video retrieval) based on visual content only.

In content-based visual retrieval, often the query-by-example paradigm (QBE) is used. A user presents one or more example documents that represent the information need and the system is supposed to return similar documents. Content-based visual retrieval systems represent queries and documents as *feature vectors* that capture one or more aspects of a document, like for example, colour, edges, texture, shape, or spatial-layout. Similarity of collection images to the query is typically measured by calculating the distance be-

tween the query and document vectors. Thus, content-based visual retrieval systems implement variants of the vector space model (Section 2.1.2). The main difference is in the features that span the space: textual terms in text retrieval and descriptions of visual aspects, or features, for content-based visual retrieval.

The remainder of this section discusses techniques in content-based visual retrieval that are related to the work presented in this dissertation. Section 2.3.1 discusses other approaches that take inspirations from text retrieval and Section 2.3.2 reviews probabilistic models for multimedia retrieval. For a general overview of the field of content-based visual retrieval the interested reader is referred to (Smeulders et al., 2000; Marques and Furht, 2002; Rui et al., 1999). Descriptions of some well-known visual retrieval systems can be found in (Ogle and Stonebraker, 1995; Pentland et al., 1996; Flickner et al., 1997; Smith and Chang, 1997).

2.3.1 Inspiration from text retrieval

One of the main research questions in the present work is to identify parallels with an approach to text retrieval. This section reviews other multimedia retrieval techniques that have been inspired by text retrieval. First, document representations that are inspired by text retrieval are discussed and then the use of relevance feedback.

Text-inspired document representations

As noted above, most content-based visual information retrieval systems implement a vector-space like model: a similarity metric, or distance in a feature-space is used to compute the RSV. A difference between the textual and visual vector spaces is that the textual space is of much higher dimensionality and more sparsely populated. In text retrieval, each of the terms in the vocabulary is a separate dimension and document representations have only non-zero values for the terms contained in the document. Thus a document lies in the subspace spanned by the terms it contains. In visual retrieval, the vector space is spanned by the features that are extracted from the documents. Typically, a visual document gets non-zero values for each feature. This means that in content-based visual retrieval the whole feature space needs to be searched, while in text retrieval often search is restricted to the subspace spanned by the query terms.

Squire et al. (1999) adopt an approach to image retrieval that is inspired by the subspace search in text retrieval. They extract colour and texture features from the images in the database and quantise each of the features,

thus obtaining a discrete set of tens of thousands of visual 'terms' that can be either present (one or more times) or absent in a document, very much like the presence or absence of terms in text retrieval. Squire et al. employ a *tf.idf*-based weighting scheme and use weighted histogram intersection to measure similarity.

Zhu et al. (2002) follow a similar approach. Their vocabulary consists of so-called *key-blocks*, the centroids of clusters of similar blocks of pixels. Like keywords in text retrieval, key-blocks are meant to represent the document content. A document is represented as a set of key-blocks by mapping each of the pixel blocks to the nearest key-block. Several clustering algorithms are proposed for generating the vocabulary of key-blocks. Zhu et al. also experiment with uni-block, bi-block and tri-block models, where blocks are respectively independent, dependent on one neighbouring block, or dependent on two neighbours (cf. n-grams for text retrieval (Jurafsky and Martin, 2000); see also Section 3.3.3). As in text retrieval, no clear improvement over uni-blocks was found.

Relevance feedback

A second text retrieval technique that has been widely applied in image retrieval is *relevance feedback* (Rocchio, 1971; Salton and Buckley, 1990). Relevance feedback is the technique to improve retrieval quality by updating the query based on relevance information provided by the user. The main idea is that terms that are frequent in relevant documents, but infrequent in irrelevant documents, are good terms to add to the query. Most content-based visual retrieval employs relevance feedback from the start. In the QBE paradigm, the initial example provided by the user can be seen as a relevance judgement. Often, content-based visual retrieval systems are interactive and users can provide feedback on the presented results in each iteration. The system updates the query representation based on the user's feedback and shows a new set of documents in the next iteration.

In the absence of manual relevance judgements, text retrieval systems have employed pseudo relevance feedback[7]. Here, the top ranked documents are assumed to be relevant and relevance feedback is applied as if these documents were manually selected by a user. The same technique is applied to multimedia retrieval by Yan et al. (2003). However, instead of using pseudo relevance feedback to learn properties of relevant documents, they use it to learn about irrelevant documents. After an initial run, they use the lowest scoring documents as negative training material for their classifiers.

[7] Also know as blind relevance feedback.

An extended version of the technique is called *maximal marginal irrelevance* (Hauptmann et al., 2003). It aims at selecting low scoring documents of maximal diversity, instead of simply taking the lowest scoring documents. This approach is inspired by another text retrieval technique called *maximal marginal relevance* (Carbonell and Goldstein, 1998). The latter tries to reduce redundancy in the ranking and aims at retrieving documents that are both relevant and novel, i.e., different from what is already found.

2.3.2 Probabilistic multimedia retrieval

Many content-based retrieval tasks can be seen as decision theory tasks. 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 can be seen as a decision theory problem: either we can classify documents 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 decision theoretic approach seems natural: an image is assigned to the class with the lowest risk, where risk is a product of the probability that the observation belongs to that class and the cost of falsely assigning an observation to that class. Most approaches however, deviate from this framework by assuming uniform cost and by equating the probability of belonging to a class to similarity in a vector space. This way, each observation is assigned to the closest class in the vector space. This section discusses approaches that stay closer to the decision theoretic framework and use probabilistic modelling at some stage. The focus is on generative modelling, as that is the approach employed in this thesis. We start with distinguishing between generative and discriminative models.

Generative models versus discriminative models

Two general approaches to classification exist: generative models and discriminative models. Comparisons of theses can be found in (Rubinstein and Hastie, 1997; Ng and Jordan, 2002). In generative models, the probability density of each of the classes, $\mathrm{p}(\cdot|c_i)$, is estimated. To classify a new observation x, Bayesian inversion is used to find the most likely class c^*:

$$c^* = \arg\max_c \mathrm{P}(c|x) = \arg\max_c \frac{\mathrm{p}(x|c)\mathrm{P}(c)}{\mathrm{p}(x)}. \tag{2.1}$$

Discriminative approaches do not try to model the class conditional densities. Instead they attempt to directly predict the most likely class given the data ($\mathrm{P}(c|x)$). The focus is on finding the class boundaries in the data space.

One good reason to use discriminative models is formulated by Vapnik (1998)[8]:

> One should solve the problem directly and never solve a more general problem as an intermediate step.

However, discriminative models completely ignore the prior information $\mathrm{P}(c)$ that may be available. Also, they require the consideration of all classes simultaneously, which may cause computational problems when the number of classes grows. In information retrieval, where the number of classes is potentially as large as the number of documents in a collection, generative models are a better solution.[9] The class conditional densities can be estimated for each of the classes (documents) separately. Together with the prior probabilities for the different classes, these conditional densities can be used to classify observations (cf. Equation 2.1).

Generative models for visual retrieval

Several ways of using generative models have been proposed for retrieval from visual collections. This paragraph reviews the proposed approaches.

The Blobworld system (Carson et al., 2002) uses the Expectation-Maximisation (EM) algorithm (Dempster et al., 1977) to find regions of similar colour and texture and position, where each region is described by a Gaussian density in the colour-texture-position domain. Their approach to finding regions is very similar to the one taken in this thesis for describing images (see Chapter 3), but once they find a clustering of pixels in the images, they throw away the probabilistic descriptions and compute colour histograms from the extracted regions. These histograms are then compared using a quadratic distance in the vector space spanned by the bins.

Hoiem et al. (2003) develop a Bayesian classifier for object-based image retrieval. The goal in their work is to identify objects as sub-images in an image collection based on a small number of examples. They use a sliding window over each image at different scales and consider each position as a possible sub-image. For each sub-image a feature vector f is computed and the probability of belonging to the class c of the target object is computed using Bayesian inversion: $\mathrm{P}(c|f) = \mathrm{p}(f|c)\mathrm{P}(c)/\mathrm{p}(f)$. The class conditional

[8] Recited from (Ng and Jordan, 2002).

[9] For the two class variant of the problem (relevant versus non-relevant), usually too little data from the relevant class is available.

densities $p(f|c)$ are estimated from the examples provided by the user. Estimation of class conditional densities based on user provided examples is similar to the document generation variant described in Chapter 5. The unconditional density $p(f)$ is estimated from a representative sample taken from the collection under study, or from a comparable collection. Class prior $P(c)$ is an unknown constant irrelevant for ranking feature vectors. To avoid overfitting in the estimation of the class conditional densities from few training examples, the authors generate additional training examples from the ones provided by the user by translating and scaling the originals. Also, they hope to reduce over-fitting by incorporating prior information about the feature distribution into the training process. This information is used in an ad hoc manner by adding some probability mass to the a priori most likely bins.

A similar approach is used by Fergus et al. (2003) for object recognition. They estimate generative models from training data for objects such as faces, motorbikes and aeroplanes using the following process. First salient regions are detected on the training images, and for each of the detected regions descriptions of its appearance, location and scale are stored. An object class is then modelled as a set of salient regions, each described as a Gaussian distribution over appearance, location and scale. Since the number of salient regions detected in an image is typically larger than the number of parts describing an object, Fergus et al. introduce an indicator variable to assign regions to parts (not all regions need to be assigned). The models are estimated using EM. The unassigned regions are used to estimate background models from. To detect objects in unseen images, they again detect salient regions and compute their feature vectors. If the likelihood ratio of being from the class model or the background model exceeds a threshold, the image is classified as containing the object. In principle, both in training and in retrieval all possible assignments of salient regions to object parts should be considered, but the authors use efficient search methods to speed up the process. Fergus et al. are able to correctly identify large proportions of the tested classes, but they also find a large number of false positives.

Schmid (2004) uses a k-means approach to find k clusters within a set of feature vectors extracted from images in a given category (zebras, cheetahs, giraffes, faces). Each cluster is then described by a Gaussian distribution. Thus, effectively, each category is described by a Gaussian mixture model, but the model is estimated using k-means, rather than on a probabilistic basis.

Several research groups have proposed to use Gaussian mixture densities to model visual information (Vasconcelos and Lippman, 1998; Greenspan et al., 2001; Luo et al., 2003). Gaussian mixture models are also used in the present work and are discussed in detail in Chapter 3. Here we briefly

discuss how they have been used up till now. Both Vasconcelos and Lippman (1998) and Greenspan et al. (2001) 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 ranked using a measure of similarity between the query and document models. Vasconcelos and Lippman (1998) 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 (Vasconcelos, 2000). Chapter 5 shows that in generic collections, the assumptions underlying the approximation may be violated and retrieval results may be sub-optimal. Greenspan et al. extend their image model to one for video retrieval by incorporating a temporal dimension in their feature space (Greenspan et al., 2002, 2004). Luo et al. (2003) 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.

2.4 Combining textual and visual approaches

The previous sections discussed text based and content-based multimedia retrieval approaches. Often, the two ways of disclosing multimedia collections are combined, but most of these combinations do not go beyond providing two different modes of disclosure, allowing the users to search either modality. Sometimes, both modalities can be searched simultaneously and the scores are combined in an ad hoc fashion. Rarely the modalities are more tightly integrated. This section reviews some exceptions.

In the ImageRover system, Sclaroff et al. (1999) integrate textual and visual information in a single feature vector to describe the multimedia documents in their collection. They use latent semantic indexing on the textual part to reduce the dimensionality of the vector space spanned by the terms in the vocabulary. For representing the visual information, they compute colour and texture descriptors, and reduce the dimensionality of the resulting feature space via principal component analysis. They compute distances in the multi-modal space as a linear combination of the distances in visual and textual subspaces.

A similar dimension reduction strategy is applied in (Westerveld, 2000). Here textual and visual features are combined in a *single* space on which dimension reduction is applied. The resulting low-dimensional space is thus a multi-modal one and reveals relations between textual and visual features.

The same approach is used by Van Gemert (2003).

Duygulu et al. (2002) treat object recognition as machine translation. They identify salient regions within images, compute feature descriptions for them and quantise the feature space. This way, they obtain a finite, discrete set of region descriptors. The EM algorithm is then used to learn a translation lexicon between this visual vocabulary and the terms from the annotations. An accompanying paper discusses a related translation approach using continuous features (Barnard et al., 2002).

Blei and Jordan (2003) have developed generative models for representing images and captions simultaneously. The visual part of their data is modelled as a mixture of Gaussian distributions, the textual part as a multinomial model. Blei and Jordan investigate several ways of coupling the two modalities. Their models are instances of the latent Dirichlet allocation (LDA) models, which are related to the approach in this thesis (see Chapter 5 for details).

Jeon et al. (2003) develop multi-modal models using techniques from cross-language retrieval. They use the vocabulary of blobs from (Duygulu et al., 2002) as discrete descriptions of images and then use language models for describing both textual and blob distributions. To relate textual and visual data for cross-modal retrieval or annotation, they compute the conditional probability of words given blobs or vice versa by marginalising over a set of annotated images. This is similar to the multi-modal approaches discussed in Chapter 5. The main difference is that they work with discrete representations of the visual data, while Chapter 5 uses continuous representations. In a follow-up paper, Lavrenko et al. (2004) extend their models for continuous descriptions of the visual information, but this is done in an ad hoc manner. They simply place spherical Gaussian distributions around the means of the region descriptors in the feature space.

2.5 Evaluating multimedia retrieval

The first four sections of this chapter have reviewed approaches to information retrieval in general and multimedia retrieval in particular that have been proposed in the literature. This section takes a different perspective and introduces methodology for evaluating different approaches.

Experimentation is an important aspect of the work presented in this book. Throughout this thesis, the proposed techniques are tested and evaluated using a number of common evaluation measures and datasets. We follow the text retrieval tradition of laboratory style testing, and measure the quality of the system ranking in a controlled setting rather than user satisfaction

in an operational setting. Section 2.5.1 discusses the laboratory setup and the nature of the test collections. Section 2.5.2 discusses the quality measures used and Section 2.5.3 discusses the relevant measures for statistical significance. A more in depth discussion of laboratory testing for multimedia retrieval can be found in Chapter 6.

2.5.1 Test collections

Laboratory tests use a *test collection* consisting of a set of documents, a set of topics and a set of relevance judgements. The documents are the basic elements to retrieve, the topics are descriptions of the information needs and the relevance judgements list the set of relevant documents for each topic.

This section introduces the two test collections used throughout this thesis: Corel and Trecvid. As will become clear in Chapter 6 Corel is more suited for system-oriented evaluation ('Does the system work well?'), while Trecvid is useful for task-oriented tests ('Is the system useful in a realistic setting?'). Both collections are used in the subsequent chapters to illustrate and evaluate model variants and parameter settings.

Corel

The Corel set is a collection of stock photographs, divided into subsets of images each relating to a specific theme (e.g., *tigers*, *sunsets*, or *English pub signs*). In a large number of publications in the field of content-based image retrieval, this image collection is used to evaluate retrieval results or to illustrate the effectiveness of a given retrieval method (e.g., Blei and Jordan, 2003; Jeon et al., 2003; Duygulu et al., 2002; Barnard et al., 2003; Belongie et al., 1998; Vasconcelos and Lippman, 2000; Li and Wang, 2003). It is important to notice that there exists no single Corel set. In fact different publications use different subsets of the totally available amount of Corel images.

The usual approach to using the Corel data set is to use the division into themes as relevance judgements. When an image from the collection is used as a query, the assumption is that an image is relevant if and only if it belongs to the same theme. Although this assumption is not always valid (see Chapter 6 for details), it is useful for system-oriented evaluation and it has been used for the Corel experiments in this work .

Trecvid

The Text Retrieval Conferences (TREC) are a series of workshops aimed at large-scale testing of information retrieval technology (Voorhees and Harman, 2002; Voorhees and Buckland, 2003). TREC has always had different tasks for different types of retrieval. These tasks are called *tracks*. In 2001, a video track (TRECVID) has started (Over and Taban, 2002; Smeaton and Over, 2003; Smeaton et al., 2003a). This track defines three tasks: shot boundary detection, feature detection and general information search. The goal of the shot boundary task is to identify shot boundaries in a given video clip. In the feature detection task, one has to assign a set of predefined features to a shot, e.g. *indoor*,*outdoor*, *people* and *speech*. In this dissertation we focus on the search task, where the goal is to find as many relevant shots as possible given a topic, a description of an information need. Topics consist of a short textual description and one or more still images or video examples. Figure 2.1 shows an example topic. The TRECVID search task distinguishes between interactive approaches, in which a user can interact with a retrieval system to locate relevant shots, and manual approaches, in which a user has one go at creating a query from a topic description and then submits this query to the system to retrieve relevant shots. Since we are interested in how well a multimedia retrieval system can work with a minimum amount of user effort, this book concentrates on manual approaches.

VT0104: Find shots of an airplane taking off

Figure 2.1: Example Multimedia Topic

Participants in TRECVID submit their top N results for each topic. The top $K < N$ results from each submission are manually judged, resulting in a set of relevance judgements for each topic[10], which together with the collection and the topics, can be used as a test collection.

[10]Everything that is not retrieved within *any* top K is assumed to be irrelevant. Consequences of this assumption are discussed in Chapter 6.

2.5.2 Measures

The goal of an information retrieval system is to present a set of relevant documents to the user. Relevant documents are those documents that satisfy the user's information need. The concepts of *recall* and *precision* are central in most evaluation measures. Recall is the fraction of relevant documents that is retrieved. Precision is the fraction of retrieved documents that is relevant.

$$\text{recall} = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents in collection}}$$

$$\text{precision} = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved}}$$

Both recall and precision are set-based measures; they operate on a fixed set of retrieved documents. Usually, as the set of retrieved documents gets larger, recall grows at the cost of precision. For evaluating *ranked lists* of results, it is common to measure precision at different recall levels. One way to do this is by measuring precision (and recall) at fixed ranks. For example after 10, 20 and 30 retrieved documents (P@10, P@20 and P@30). Alternatively, precision can be measured after each relevant document that is retrieved. Suppose four relevant documents exist in the collection for a given information need and three of them are retrieved at ranks 1, 4 and 7. Then the precision values at these levels are respectively $P@1 = 1$ ($\frac{1}{1}$), $P@4 = 0.5$ ($\frac{2}{4}$) and $P@7 = 0.43$ ($\frac{3}{7}$). The average of the precision values after each relevant document that is retrieved is called the *(non-interpolated) average precision*. To calculate it, the precision value of relevant documents that are not retrieved is assumed to be 0, thus the average precision in the given example is $(1 + 0.5 + 0.43)/4 = 0.48$.

It is good practise to evaluate information retrieval systems on more than a single query (or information need) and to report averages over all queries. A commonly used single measure is the *mean average precision* (MAP), the mean of the average precision values over all topics in the evaluation set. Another way of presenting multiple query results is with a *recall-precision graph*. A recall-precision graph plots precision against recall and is constructed by computing interpolated precision at 11 fixed recall points (0 to 1 in steps of 0.1). The interpolated precision at recall level l is defined as the maximum precision for any recall level greater than l. A graph for multiple queries is constructed by averaging the 11-point precision values over all queries. Figure 2.2 shows the (single query) interpolated recall-precision graph for the example introduced above. Figure 2.3 shows an example recall-precision graph constructed from multiple queries.

MAP and recall-precision graphs are the most commonly used measures. They are used in the experiments in the following chapters.

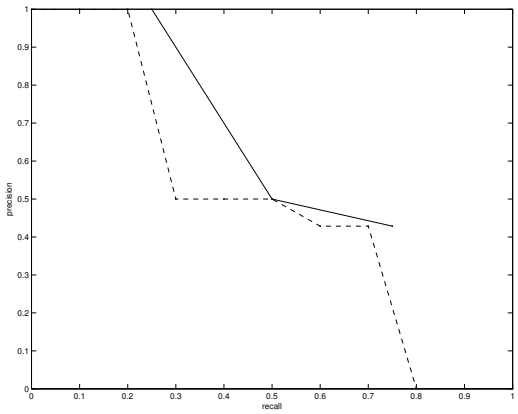


Figure 2.2: Interpolated Recall example. Three relevant documents retrieved at ranks 1, 4 and 7, fourth not retrieved. Exact recall levels: 0.25, 0.5, 0.75 and 1.0; precision at these levels (solid line): 1.0, 0.5, 0.43 and 0.0. Interpolated precision (dashed line) for standard recall levels 0, 0.1 and 0.2 is 1, for 0.3, 0.4 and 0.5 is 0.5, for 0.6 and 0.7 is 0.43 and for 0.8 or greater 0.0

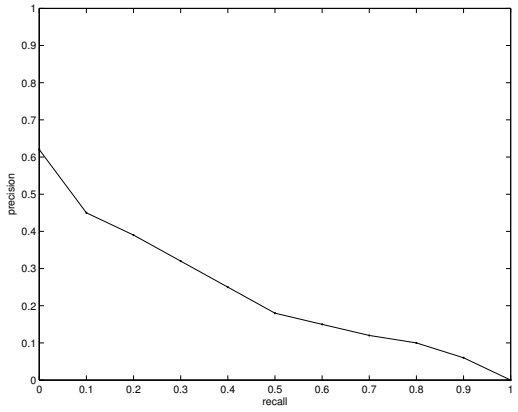


Figure 2.3: Example Recall-Precision Graph

2.5.3 Statistical significance

In a comparison between two systems or system variants on a test collection, simply stating the absolute or relative difference in performance does not tell everything. It is important to distinguish between differences that are due to chance and differences that are due to the fact that one system is better than the other. To test this, significance tests have been developed. These statistical tests compare results for two runs and decide whether the variation in scores are due to a difference between the systems or due to chance.

The non-parametric Wilcoxon signed-ranks test is often used to test for significance on the outcomes of information retrieval experiments. Zobel (1998) finds that the Wilcoxon test offers more reliability and greater discriminative power than its alternatives. This test is used in subsequent chapters.

For each pair of observations (e.g., average precision scores for a given query for different approaches) the Wilcoxon signed-rank test computes the absolute difference. Then these differences are ordered and each difference is replaced by its rank. Subsequently, the sums of ranks corresponding to positive differences and ranks corresponding to negative differences are computed. The idea is that if the two approaches do not differ significantly, neither will these sums. Thus, if a large enough difference in the sums is observed, one approach can be considered significantly better than the other.

2.6 Discussion

This chapter has introduced some of the most important information retrieval models and discussed issues that are closely related to the work presented here, viz., text retrieval for multimedia collections, content-based visual retrieval (with a focus on probabilistic techniques), combining textual and visual approaches and evaluating multimedia retrieval.

The work presented here stems from the generative modelling approaches to text retrieval and content-based image retrieval introduced in Sections 2.1 and 2.3.2. These models provide a generic way of modelling data that can be applied in a broad domain with little user effort. The contribution of our work is three-fold. First, we bring together generative models for text retrieval and explore possibilities for cross-fertilisation. Second, by using the same probabilistic basis for both modalities a seamless integration of textual and visual evidence becomes possible. Third, we evaluate the proposed models on the task of ad hoc retrieval from heterogeneous multimedia archives.

The next chapter introduces the generative probabilistic models in detail.

3

Generative probabilistic models

This chapter introduces the models that are the basis of the research presented in this thesis. The chapter starts with an analogy which should give the reader an intuition of the nature of the models and their use for information retrieval (Section 3.1). Section 3.2 introduces the notation used throughout this thesis. The generative models are formally introduced in Section 3.3. Section 3.4 shows how the models can be used for information retrieval and Section 3.5 explains the basics of estimating the model parameters. In Section 3.6, we see the first bit of cross-fertilisation where ideas from text retrieval for improving the parameter estimates are transferred to the visual domain. The chapter concludes with a section that places the generative models in perspective (Section 3.7). The models presented in this chapter have been discussed before in (Westerveld et al., 2003b; Westerveld, 2002).

3.1 Introduction

Our approach to image retrieval is similar to ordering pieces of a jigsaw puzzle. Suppose we have been solving jigsaw puzzles all weekend and put all puzzles in their boxes again on Sunday evening. Now it is Monday morning and while cleaning the room, we find a forgotten piece of one of the jigsaws. Of course, in practise, we would keep the piece separate until we solve one of the puzzles again and discover that a piece is missing. But suppose now that we have to make a decision and put the piece in one of the boxes. To put it in the proper box, we have to guess to which puzzle this piece belongs. The only clues we have are the appearance of the piece at hand and our memory of the puzzles we solved. A good solution would be to put the piece in the box to which it most likely belongs given these clues. If for example, the

piece at hand is mainly blue with a watery texture, it is most likely to come from a jigsaw with a lot of water.

In our retrieval framework, instead of boxes with jigsaws we have a collection of documents and instead of a forgotten jigsaw piece, we have a query. The goal now, is to find the document that is most likely given the query, similar to choosing the most likely box to put the jigsaw piece in. Although at a glance jigsaws seem to be analogous to image retrieval only, the quest for the source of a piece of information is of course applicable to any information retrieval domain. This generative approach to information retrieval –find the generating source of a piece of information– has proved successful in media specific tasks, like language modelling for text retrieval (Ponte and Croft, 1998; Hiemstra, 1998) and Gaussian mixture modelling for image retrieval (Vasconcelos, 2000; Greenspan et al., 2001; Luo et al., 2003).

3.2 Notation and terminology

This section introduces the basic terminology used in the present work. Standard conventions are used for representing vectors, matrices and sets (See Appendix A). Throughout this thesis, the term *document* is used to refer to the basic entity to retrieve. Documents can be images or pieces of text, but in general we will assume multimodal documents. The term *bag* is used for unordered collections of elements that may contain duplicates (bags are also known as multisets).

Figure 3.2 and 3.3 illustrate the representations for visual and textual documents, using the example image and caption shown in Figure 3.1. A visual representation is constructed by computing a feature vector for each square block of n by n pixels. The vector describes the colour and texture information in the corresponding pixel block as well as the position of the block (details are deferred to the next chapter). Both the blocks and the feature vectors representing them will be referred to as *samples*. Samples can be compared to the pieces of the jigsaw puzzle from the introductory example. A representation of a textual document is constructed by counting the number of occurrences for each of the terms in the vocabulary and representing them as a vector.[1]

More formally, we define the following terms:

- A collection is a set of multimodal documents: $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_M\}$.
- A multimodal document $\mathcal{D}$ is a tuple of a textual document $\mathcal{T}$ and visual document $\mathcal{V}$: $\mathcal{D} = (\mathcal{T}, \mathcal{V})$.

[1]The position in the vector serves as an index into the vocabulary.

Pt. Reyes National Seashore – Tule Elk.

Figure 3.1: Example image with caption.

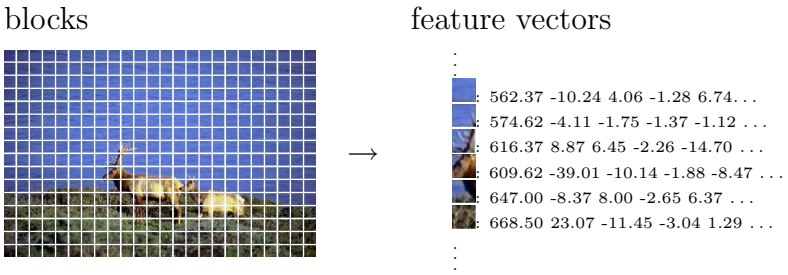


Figure 3.2: Illustration of visual document representation.

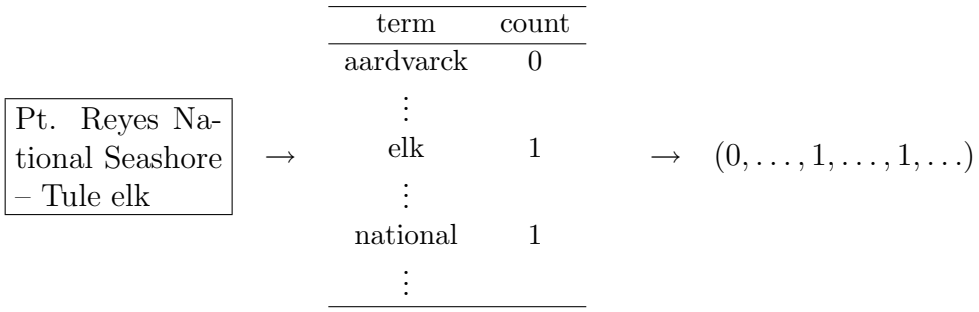


Figure 3.3: Illustration of textual document representation

- A textual document is a bag of terms: $\mathcal{T} = \{\text{term}_1, \text{term}_2, \ldots, \text{term}_N\}$. Alternatively, a textual document is represented as a vector of term counts: $\boldsymbol{t} = (t_1, t_2, \ldots, t_T)$, where T is the number of terms in the vocabulary.

- A visual document $\mathcal{V}$ is composed of a number of small, square blocks of pixels, each of them represented by a feature vector. In the following, the term *sample* refers to both a pixel block and the feature vector describing it. Thus, $\mathcal{V}$ is represented as a bag of S visual samples: $\mathcal{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_S}\}$.

- Each of these samples is a n-dimensional feature vector: $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$. The feature vectors describe colour, texture and position of the pixel block. Details are described in Chapter 4.

Note that each textual document is represented as a single vector, or as a single point in the space spanned by the terms in the vocabulary, whereas visual documents are represented as a cloud of points in the feature space; one point for each sample. This is mere convention rather than something inherent to the way of modelling. Usually, textual document statistics are computed for full documents (e.g., term frequencies), while image statistics are often reported for smaller regions in an image (e.g., local colour histograms).

3.3 Generative probabilistic models

Since the goal in information retrieval is to find the best document given a query, one could decide to model the probability of a document given a query directly. In the jigsaw example, this would mean that a direct mapping from an appearance of a piece to a jigsaw box is needed (P(box|piece)). This way of modelling the problem is known in the classification literature as *discriminative classification* (see also Section 2.3.2. In some cases, for example when there are many different boxes, it is hard to learn this direct mapping. In such cases, it is useful to apply Bayesian inversion and estimate for each box the probability that this box produced the piece at hand (P(piece|box)). This approach is known as *generative classification*. In this approach, each box has a model of the type of pieces it generates. The probability of generating the jigsaw piece at hand is computed for each model and that probability is used to find the most likely box. Section 3.7 discusses the relationship between classification and retrieval in more detail. In this

section, the basic explanation of the generative models is continued. In information retrieval, many possible sources for a query exist; each document in a collection can be a source. Therefore, learning a discriminative classifier is hard and a generative approach is a natural way of modelling information retrieval. It is important to realise that in such an approach, a separate distribution is estimated for each of the documents in the collection. One of the nice things about generative probabilistic models is that they can easily be understood without digging into the details of estimating the models' parameters. Therefore, in the remainder of this section parameter estimation is put aside and only the basics of the models are explained. This section starts with some examples of generative models (Section 3.3.1). Section 3.3.2 specialises to generative image models and Section 3.3.3 discusses generative language models.

3.3.1 Examples

Generative probabilistic models are random sources that can generate (infinite) sequences of samples according to some probability distribution (see for example Duda et al., 2000). In the simplest case, the model generates samples independently, thus the probability of a particular sample is independent of the samples generated previously. These simple models, often called memoryless models, will be the primary focus in this thesis. A good example of a generative source with a memoryless model is an ordinary dice. The model describes the process of throwing the dice and and observing the outcome. If the dice is fair, throwing it generates positive integers between 1 and 6 according to a uniform distribution:[2]

$$\mathrm{P}(i) = \frac{1}{6}, \text{for } i \in \{1, 2, 3, 4, 5, 6\}. \tag{3.1}$$

In a memoryless model, the observations or samples are assumed to be independent, so the probability of observing a particular sequence is calculated as the product of the probabilities of the individual observations.

$$\mathrm{P}(\{i_1, i_2, \ldots, i_n\}) = \prod_{j=1}^{n} \mathrm{P}(i_j) \tag{3.2}$$

Section 3.4 returns to calculating the probabilities of observations. Here, the focus is on the probabilistic models themselves. A probabilistic model is

[2]Throughout this thesis, random variables are omitted from the notation of probability functions, unless this causes confusion. Thus, $\mathrm{P}(i)$ means the probability that the random variable describing the observed outcome from throwing the dice takes value i.

an abstraction from the physical process that generates the data. Instead of specifying that the sequence of positive integers is produced by throwing an ordinary fair dice, it suffices to state that there is some source that generates integers between 1 and 6 according to a uniform distribution (Equation 3.1). The underlying physical process can remain unknown. Still, to understand the models it is often useful to think of simple processes like throwing a dice, drawing coloured balls from an urn, or drawing jigsaw pieces from a box.

Generative models can also be more complicated and have a hierarchical structure. Suppose we have two dice: Dice A, with the usual faces 1 through 6 ($A = (1, 2, 3, 4, 5, 6)$)[3] and dice B, which has ones on all faces ($B = (1, 1, 1, 1, 1, 1)$). Now we can imagine the following random process:

1. Pick a dice according to a uniform distribution.
2. Sample a number by throwing the chosen dice.

For this generative process, the probability of observing a single sample i is:

$$\mathrm{P}(i) = \mathrm{P}(A) \cdot \mathrm{P}(i|A) + \mathrm{P}(B) \cdot \mathrm{P}(i|B) = \begin{cases} \frac{1}{2} \cdot \frac{1}{6} + \frac{1}{2} \cdot 0, & \text{for } i \in \{2, 3, 4, 5, 6\} \\ \frac{1}{2} \cdot \frac{1}{6} + \frac{1}{2} \cdot 1, & \text{for } i = 1 \end{cases} \tag{3.3}$$

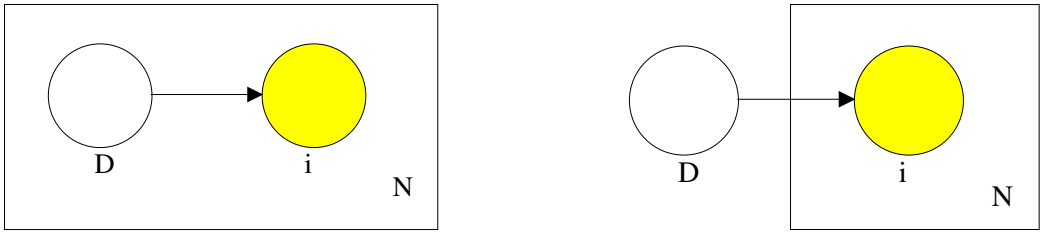
A generative process with a model like this is called a *mixture model*. It is a weighted sum of a number of different probability distributions. As will become clear in Section 3.3.2, mixture models are useful for describing the mixture of aspects that can be present in images.

It is often insightful to represent generative models in a graphical manner. For graphical representations, we follow the standards described in (Jordan, 2003), where random variables are represented as nodes and dependencies between them as edges. Observed variables are represented as solid nodes and hidden, or unobserved, variables as open nodes. A box or plate around a part of the graph indicates repetition, i.e., the repeated sampling of variables. As an example, Figure 3.4 represents two variants of drawing a sequence of N numbers from the hierarchical dice. The variant on the left represents the process as described above: for each number, we pick a new dice. The variant on the right represents the case where we select a dice once for the whole process and then repeatedly sample numbers by throwing that dice.

3.3.2 Generative image models

As stated in the introduction to this chapter, generative image models are like the boxes of jigsaw puzzles, from which one can randomly draw pieces.

[3] We represent a dice as a list of faces.



Sampling with repeated choice of dice Sampling from a single dice

Figure 3.4: Graphical representations for dice example variants.

An important difference though is the following. Jigsaw boxes contain a finite number (say 1000) of discrete pieces; a piece is either in there or not. By sampling from the box *with replacement*, we can draw infinitely many pieces, but each piece has to be one of the fixed set of 1000 pieces. The generative image models however, are probability distributions over a (high dimensional) *continuous* feature space. The number of different samples that can be drawn is infinite. The models describe where in the feature space, we are most likely to observe samples and what kind of variance can be expected. The nature of the feature space, i.e., the set of features used for describing a sample, is discussed in Section 4.1.2. Here, a sample $\boldsymbol{v}$ is assumed to be described by a N-dimensional feature vector $\boldsymbol{v} = (v_1, \ldots, v_N)$.

Gaussian mixture models

Normal distributions, or Gaussian distributions as they are often called, are appropriate models for the situation in which there exists an ideal point in a feature space and all observations are assumed to be versions of this ideal feature vector that are randomly corrupted by many independent small influences (Duda et al., 2000). For simple images this is the case, one can easily imagine a single ideal point in feature space describing for example the perfect water texture. All observations from that water class can be seen as versions of the ideal water texture that have been corrupted by many independent causes (lightning condition, camera angle, etc.).

However, most real-life images show more than a single texture or object. Therefore, Vasconcelos proposes to use a Gaussian mixture model for modelling images with multiple colours and textures (Vasconcelos, 2000). Mixtures of Gaussian distributions are popular densities for modelling all sorts of random sources (Everitt and Hand, 1981; Titterington et al., 1985). In principle, any function can be approximated by a mixture of a large enough number of Gaussians (McLachlan and Peel, 2000). The more components in the mixture, the better the approximation.

A finite mixture density is a weighted sum of a finite number (C) of density functions (e.g., Titterington et al., 1985; Duda et al., 2000):

$$\mathrm{p}(x) = \sum_{i=1}^{C} \mathrm{P}(c_i)\mathrm{p}(x|c_i). \tag{3.4}$$

The mixing weights $\mathrm{P}(c_i)$ are the prior probabilities of the components c_i in the mixture.

Titterington et al. (1985) divide the usage of mixture models in two broad classes: *direct application* and *indirect application*. Direct application is used to refer to situations in which it is believed that there exists a number (C) of underlying categories or sources such that the observed samples all belong to one of these. Indirect application refers to a situation in which a mixture model is only used as a mathematical way of obtaining a tractable form of analysing data. Modelling images using finite mixture models is somewhere halfway on the continuum from direct to indirect application. On the one hand, the idea is that an image can contain only a finite number of things; each sample is assumed to be generated by one of the mixture components. For example, one component might describe the grass, another the water and a third the sky in an image. This is the direct application view. On the other hand, we do not explicitly model grass, water and sky. We merely believe that to model the many different facets of an image, a mixture of distributions is needed. This mixture model describes image samples without explicitly separating the components. In that sense, mixture modelling is just a mathematical tool to describe images (indirect application view). Still, the direct application view with separate components for modelling grass, water and sky, is a useful way of thinking about finite mixture models for images.

Gaussian mixture models for representing images

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 deferred to Section 3.5.1. The process of generating an image is assumed to be the following (see Figure 3.5):

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

2. For each sample $\boldsymbol{v}$ 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 $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

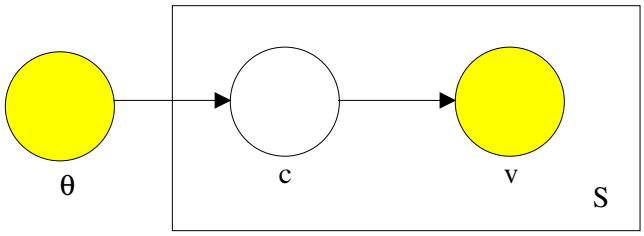


Figure 3.5: Graphical representation of Gaussian mixture model.

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 $\boldsymbol{v}$ from a Gaussian mixture model with parameters $\boldsymbol{\theta}$ is thus defined as follows.

$$\mathrm{p}(\boldsymbol{v}|\boldsymbol{\theta}) = \sum_{i=1}^{C} \mathrm{P}(c_i|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{v}|c_i, \boldsymbol{\theta}) \tag{3.5}$$

$$= \sum_{i=1}^{C} \mathrm{P}(c_i|\boldsymbol{\theta}) \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma_i}|}} e^{-\frac{1}{2}(\boldsymbol{v}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1}(\boldsymbol{v}-\boldsymbol{\mu_i})} \tag{3.6}$$

A visualisation of the model built from the image in Figure 3.1 is shown in Figure 3.6. For this example, a Gaussian mixture with three components is estimated from the set of feature vectors extracted from the image (cf. Figure 3.2).[4]. The resulting model is described by the mean vectors and covariance matrices of the three components in the high-dimensional feature space and by the prior probabilities of the components. The figure shows a projection of the components onto the two-dimensional subspace defined by the position in the image plane (i.e., the space spanned by the x and y coordinates of the feature vectors). The ellipsoids in the image plane show the mean position of the three components along with their variation. The fill areas, are the areas in the image plane, where the standard deviation

[4]The process of building a model is described in Section 3.5.1

from the mean position for a given component is below 2. The colour of the area is a representation of the component's other dimensions: it shows the mean colour and mean texture. Variance in colour and texture information are not visualised. The bars to the right of each component indicate the component's prior probability.

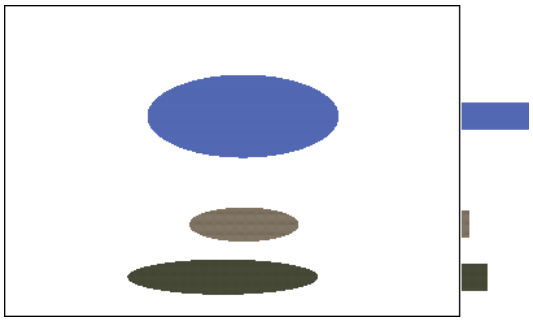


Figure 3.6: Visualisation of a model of the image in Figure 3.1.

As mentioned at the beginning of this section, any distribution can be approximated arbitrarily closely by a mixture of Gaussians. The higher the number of components in the mixture, the better the approximation can be. However, keeping in mind that the models will be used for retrieval, a perfect description of an image is not the ultimate goal. The goal is to find images that are similar to a query image. A perfect model would only be able to find exact matches and those are not the most interesting ones. Therefore, it is important to avoid over-fitting. Typically, a low number of components (between 4 and 32) will be used. Hopefully, this is enough to capture the most important aspects of an image. Chapter 4 describes experiments in which the number of components is varied to find an optimum. The next subsection describes generative models for describing text.

3.3.3 Generative language models

Since the 1970s, language models have been heavily used in speech recognition (Jelinek, 1997), but also for other natural language processing tasks (Cutting et al., 1992; Brown et al., 1990). Since 1998, generative language models have become increasingly popular in information retrieval (Kalt, 1998; Ponte and Croft, 1998; Hiemstra, 1998; Miller et al., 1999; Hiemstra, 2001; Zhai, 2002).

A language model is a probability distribution over strings of text in a given language. It simply states how likely it is to observe a given string in

a given language. For example, a language model for English should capture the fact that the term *the* is more likely to occur than the term *restaurant*. When context is taken into account this might change. For example, after seeing the phrase

They went to an Italian

restaurant is a more likely completion than *the*. In language modelling tasks like automatic speech recognition and spelling error correction, a limited amount of context is typically taken into account and so called *N-gram models* are used (Jurafsky and Martin, 2000). In N-gram models, the probability of observing a given term only depends on the previous $N-1$ terms. If bigram models ($N = 2$) are used, the probability of the next term in the example given above would only depend on *Italian*.

Unigram language model

For the present goal, retrieving relevant documents for a given query, context is of minor importance. Although language models are generative models, in retrieval they are not used to actually generate *new* pieces of text. As long as the models capture most of the topicality of a text, they are useful. Therefore, context is typically ignored in information retrieval and terms are assumed to be generated independently. The models are thus memoryless. In language modelling memoryless language models are known as unigram language models. Song and Croft experimented with higher order (N-gram) language models for information retrieval and found no significant improvement over unigram models (Song and Croft, 1999). In the rest of this thesis unigram language models will be used to model textual information.

In the unigram language modelling approach to information retrieval, documents are assumed to be multinomial sources generating terms. This multinomial basis is not mentioned explicitly in all of the above references, but it is useful to take this view because it clearly shows the generative probabilistic nature and it nicely separates the model from the estimation of the model parameters, which is discussed in Section 3.5.2.

Multinomial sources are often introduced using urns with coloured balls, but boxes with jigsaw pieces are equally suitable. Suppose we have a jigsaw puzzle box that contains pieces with grass, pieces with water and pieces with sky. Now, if we draw ten pieces from this box with replacement, what is the probability of observing exactly five grass pieces, two water pieces and three sky pieces? This can be modelled using a multinomial distribution. For unigram language modelling, instead of jigsaw pieces of a particular type (grass, water, sky), we have terms in a given language. A question could now

be: If we draw 6 terms from English, what is the probability of observing each of the terms *an*, *Italian*, *restaurant*, *they*, *to* and *went* exactly once? In the language modelling approach to information retrieval, instead of having a single model for a language, each document in a collection is modelled as a separate multinomial source. Each of these models is described by a vector of term probabilities $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_T)$.

The generative process for textual documents, as visualised in Figure 3.7 is very simple:

1. Pick the language model $\boldsymbol{\phi}$ for the document
2. For each term
   (a) Draw a random term from $\boldsymbol{\phi}$ according to the multinomial distribution mult($\boldsymbol{\phi}$)

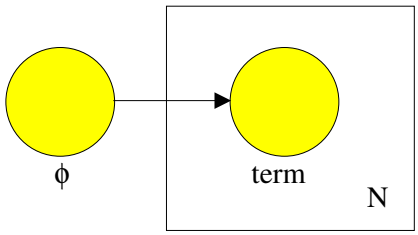


Figure 3.7: Graphical representation of language model.

Again, the model that generates the samples (terms) is an observed variable; each document has its own, known generative model $\boldsymbol{\phi}$. The probability of observing a particular document $\boldsymbol{t} = (t_1, t_2, \ldots, t_T)$, from this model is defined as:

$$\mathrm{P}(\boldsymbol{t}|\boldsymbol{\phi}) = \frac{\left(\sum_{i=1}^{T} t_i\right)!}{\prod_{i=1}^{T} t_i!} \prod_{i=1}^{T} \phi_i^{t_i} \tag{3.7}$$

The second factor in this equation ($\prod_{i=1}^{T} \phi_i^{t_i}$), is the joint probability of observing the term counts for individual terms ($\mathrm{P}(\mathrm{term}_i|\boldsymbol{\phi}) = \phi_i$). The unigram assumption states that all observations are independent, thus the joint probability is simply the product of the probabilities of the individual terms. The normalisation factor $\left(\sum_{i=1}^{T} t_i\right)!/(\prod_{i=1}^{T} t_i!)$ implements the *bag-of-words* model, it states that an observation (a query or a document) is a bag, the ordering of the terms is unimportant. A simple example will clarify this. Suppose we have a vocabulary with only 4 terms: A,B,C and D and observation

$ABAC$, then $\boldsymbol{t} = (2, 1, 1, 0)$. Note that in the representation of the observation, the order of the terms is already ignored, it simply says there are 2 A's, a B, a C and no D's. Thus, the probability of observing this $\boldsymbol{t}$ from a given model $\boldsymbol{\phi}$, is in fact the probability of drawing any permutation of the original string $ABAC$ ($\mathrm{P}(\boldsymbol{t}) = \mathrm{P}(ABAC)+\mathrm{P}(AABC)+\mathrm{P}(ABCA)+\mathrm{P}(ACAB)+\ldots$). In total $\frac{(2+1+1+0)!}{2!1!1!0!} = \frac{24}{2} = 12$ different possible permutations exist. Thus $\mathrm{P}(\boldsymbol{t}) = 12\phi_1^2\phi_2^1\phi_3^1\phi_4^0$.

3.4 Retrieval using generative models

By drawing enough pieces from a single model (or box, to take the jigsaw analogy), a random image can be generated. An example of a random image from the model visualised in Figure 3.6 is shown in Figure 3.8. Different models will produce different random images, just like different boxes can contain different jigsaws. This idea can be used to rank documents.

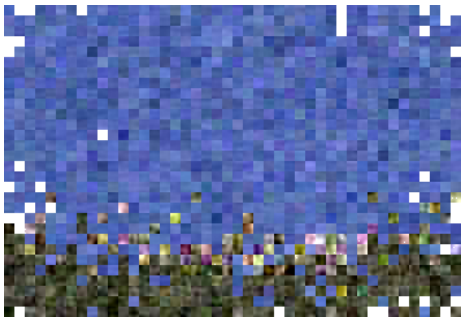


Figure 3.8: Random sample from image model presented in Figure 3.6.

The idea of ranking models based on observations is illustrated by a simple dice example. Suppose we have two dice:

$$\begin{aligned} D_1 &= (1, 2, 3, 4, 5, 6) \\ D_2 &= (1, 1, 3, 4, 5, 6) \end{aligned} \tag{3.8}$$

Say someone tells us that a sequence of 5 throws with one of them resulted in the observation: $O = (4, 3, 4, 3, 1)$. We can then easily calculate the likelihood of observing this sequence given each of the models.

$$\begin{aligned} \mathrm{P}(O|D_1) &= (\tfrac{1}{6})^5 \\ \mathrm{P}(O|D_2) &= (\tfrac{1}{6})^4 \cdot \tfrac{2}{6} \end{aligned} \tag{3.9}$$

$\mathrm{P}(O|D_2) > \mathrm{P}(O|D_1)$, thus the observation is more likely under D_2. We call this probability of an observation O given a model D the *Sample Likelihood*: it is the likelihood of observing this sample.

The same principle can be used to rank documents given a query. The assumption is that the query is an observation from one of the generative document models in the collection and the goal is to find the document model under which this query is most likely. For textual queries $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$, we can simply use Equation 3.7. For visual queries $\mathcal{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_S}\}$, assuming memoryless models, we can compute the joint likelihood of observing all samples by taking the product of the likelihoods for the individual samples $\boldsymbol{v_j}$ (Equation 3.5).

3.5 Maximum likelihood estimates

In the previous sections, the assumption has been that the model parameters ($\boldsymbol{\phi}$ and $\boldsymbol{\theta}$) are known. Given the parameters, it is straightforward to use the models for ranking documents (see Section 3.4). In general however, the parameters of a specific document model are unknown. Usually, the only available information is the representation of the documents. A common way to use this data is to assume that they are observations from the models and use them as training samples to estimate the unknown model parameters. As a first step to estimating these parameters, we will use the *maximum likelihood estimate*. This estimate is defined as the parameter setting that maximises the likelihood of the observed samples. Thus, for a set of training samples $\mathcal{S} = \{s_1, s_2, \ldots, s_K\}$ and model parameter $\boldsymbol{\psi}$, the maximum likelihood estimate $\boldsymbol{\psi}_{\mathrm{ML}}$ is defined as:

$$\boldsymbol{\psi}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\psi}} \prod_{s \in \mathcal{S}} \mathrm{P}(s|\boldsymbol{\psi}) \tag{3.10}$$

The following sections apply this approach separately to Gaussian mixture models and language models. Techniques for handling unobserved data and for improving generalisation capabilities are discussed in Section 3.6.

3.5.1 Estimating Gaussian mixture model parameters

The maximum likelihood estimate for a Gaussian mixture model from a set of samples $\mathcal{V}$ (an image) is defined as follows.

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \prod_{\boldsymbol{v} \in \mathcal{V}} \mathrm{P}(v|\boldsymbol{\theta}) = \prod_{\boldsymbol{v} \in \mathcal{V}} \sum_{i=1}^{C} \mathrm{P}(c_i|\boldsymbol{\theta}) \; \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma_i}|}} e^{-\frac{1}{2}(\boldsymbol{v}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1} (\boldsymbol{v}-\boldsymbol{\mu_i})} \tag{3.11}$$

This equation is hard to solve analytically, but we can use the Expectation Maximisation (EM) algorithm (Dempster et al., 1977) as described below.

As mentioned in Section 3.3.2, one way to look at mixture modelling for images is by assuming that an image shows a limited number of different things (such as grass, sky, water), each of which is modelled by a separate Gaussian distribution. Each sample in a document is then assumed to be generated from one of these Gaussian components. This point of view, where ultimately each sample is explained by one and only one component, is useful when estimating the parameters for a Gaussian mixture model.

To accurately describe the different components of a Gaussian mixture model for a given document, it is necessary to decide which of the document's samples are generated by which component. The assignments of samples $\boldsymbol{v_j}$ to components C_i are unknown, but they can be viewed as hidden variables and the EM algorithm can be applied. 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 (M-step).

The EM algorithm first assigns each sample to a random component. Next, the first M-step computes the parameters ($\boldsymbol{\theta_i}$) for each component, based on the samples assigned to that component. Using maximum likelihood estimates, this comes down to computing the mean and variance of the feature values over all samples assigned to the component.This assignment of samples to components is a *soft* clustering, a sample does not belong entirely to one component. In fact, we compute means, covariances and priors on the weighted feature vectors, where the feature vectors are weighted by their proportion of belonging to the class under consideration. In the next E-step, the class assignments are re-estimated, i.e., the posterior probabilities ($\mathrm{P}(c_i|\boldsymbol{v_j})$) are computed. We iterate between estimating class assignments (expectation step) and estimating class parameters (maximisation step) until the algorithm converges. Figure 3.9 is a visualisation of training a model from the image in Figure 3.1. From top to bottom, it shows alternate sample assignments (E-step) and visualisations of the intermediate models (M-step). After 10 iterations already, the model accurately distinguishes water, grass and elks.

More formally, to estimate a Gaussian mixture model from a document $\mathcal{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_S}\}$, the following steps are alternated:

E-step
Estimate the hidden assignments h_{ij} of samples to components for each sam-

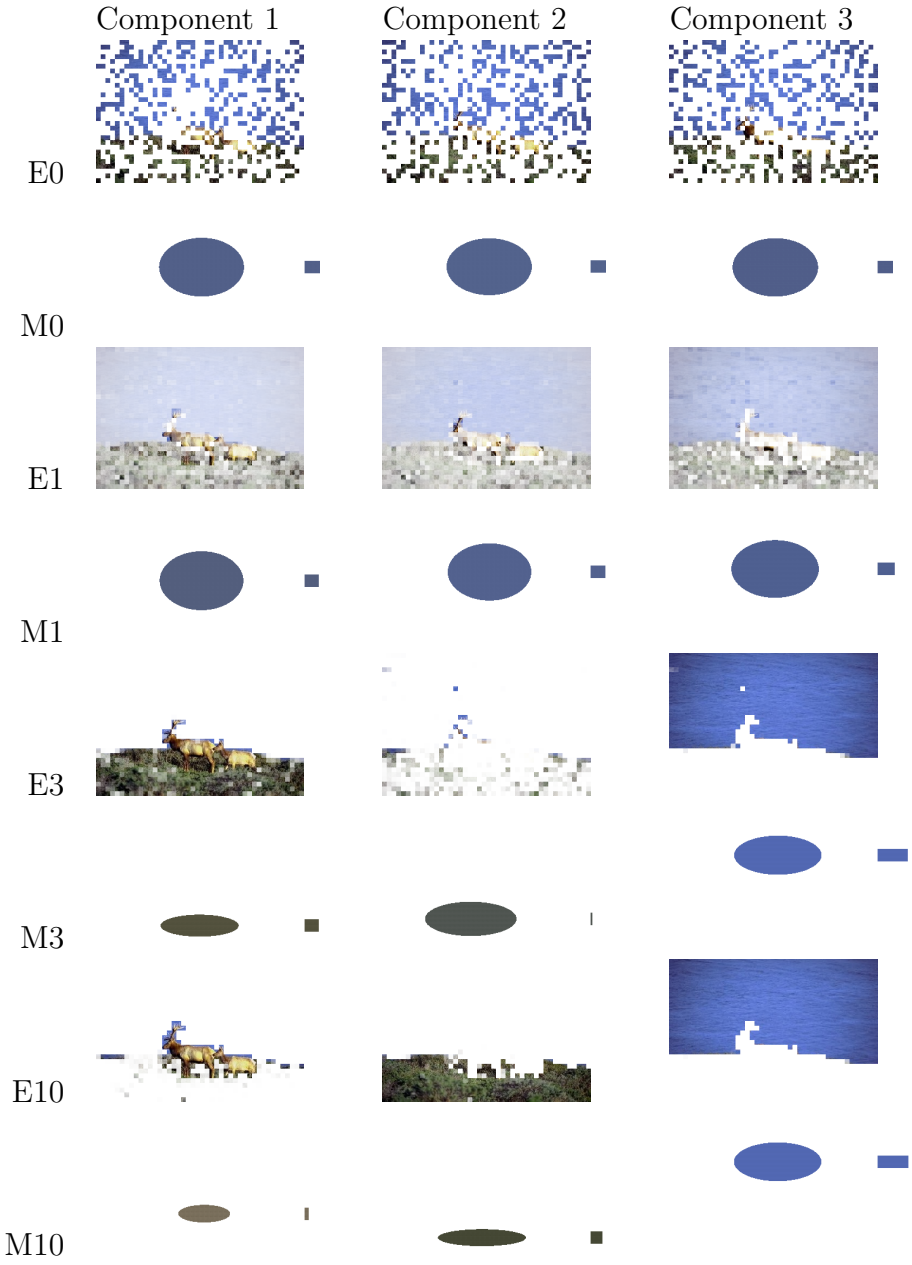


Figure 3.9: Visualisation of the estimation of parameters for a Gaussian mixture model built form the image shown in Figure 3.1. E and M steps are shown after initialisation and after 1, 3 and 10 iterations. The E-steps show to what degree each sample is assigned to each component (higher transparency indicates a lower degree of assignment). The M-steps show visualisations of the models (cf. Figure 3.6)

ple x_j and component c_i

$$h_{ij} = \mathrm{P}(c_i|\boldsymbol{v_j}) = \frac{\mathrm{p}(\boldsymbol{v_j}|c_i)\mathrm{P}(c_i)}{\sum_{c=1}^{C} \mathrm{p}(\boldsymbol{v_j}|c_c)\mathrm{P}(c_c)} \tag{3.12}$$

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

$$\boldsymbol{\mu_i}^{\text{new}} = \frac{\sum_j h_{ij}\boldsymbol{v_j}}{\sum_j h_{ij}}, \tag{3.13}$$

$$\boldsymbol{\Sigma_i}^{\text{new}} = \frac{\sum_j h_{ij}(\boldsymbol{v_j} - \boldsymbol{\mu_i}^{\text{new}})(\boldsymbol{v_j} - \boldsymbol{\mu_i}^{\text{new}})^T}{\sum_j h_{ij}}, \tag{3.14}$$

$$\mathrm{P}(c_i)^{\text{new}} = \frac{1}{N}\sum_j h_{ij} \tag{3.15}$$

The algorithm is guaranteed to converge (Dempster et al., 1977). The error after each iteration is the negative log likelihood of the training data

$$E = -\log \mathrm{p}(\mathcal{V}) = -\sum_{\boldsymbol{v}\in\mathcal{V}} \log \mathrm{p}(\boldsymbol{v}|\boldsymbol{\theta}). \tag{3.16}$$

This error will decrease with each iteration of the algorithm, until a minimum is reached. The minima are local ones. Different initialisations may lead to different results. Section 4.2.2, investigates the effects of this on retrieval results experimentally.

3.5.2 Estimating language model parameters

The maximum likelihood estimates for the parameters of the multinomial distribution for a given document are straightforward. They are simply the relative frequency of the terms in the document. If a document is represented as a vector of term counts, $\boldsymbol{t} = (t_1, t_2, \ldots, t_T)$, then ϕ_i, the probability of term i in this document, is estimated by:

$$\phi_{i_{\text{ML}}} = \frac{t_i}{\sum_{j=1}^{T} t_j}. \tag{3.17}$$

3.6 Smoothing

If maximum likelihood estimates (Equation 3.17) are used to find the language model parameters, we run into the so-called *zero-frequency problem*,

a sparse data problem. Terms that did not occur in the training data for a document are assigned zero probability ($\phi_i = 0$ for these terms). This means that a query containing such a term will get zero probability for this document model, no matter how likely the other query terms are.

Consider for example the dice example of Section 3.4, where we introduced the following two dice.

$$\begin{array}{ll} D_1 & = (1,2,3,4,5,6) \\ D_2 & = (1,1,3,4,5,6) \end{array} \tag{3.18}$$

Now, if we observe the sequence $O = (1,2,1,4,3)$, we would conclude the observation comes from D_1, since $\mathrm{P}(2|D_2) = 0$ and thus $\mathrm{P}(O|D_2) = 0$. If D_2 indeed does not have a 2 on one of its faces, this is correct, but if the distribution is estimated from data (as it is in the generative document models) it may not be. Suppose we buy a dice in a shop, we roll it six times and we observe the sequence $(1,1,3,4,5,6)$, concluding that these six observations correspond to the six faces and that there is no 2 on this dice does not seem wise.

3.6.1 Interpolation

Smoothing solves the zero-frequency problem by transferring some of the probability mass from the observed samples to the unseen samples. The specific smoothing technique used commonly in the language modelling approach to information retrieval is *interpolation*, also known as Jelinek-Mercer smoothing (Jelinek and Mercer, 1980). For multimedia material, and especially for video data, interpolation is useful, since it allows for easy extension of the language models for describing different levels of a document, like shots, scenes and videos (See Section 3.6.4). Therefore, this technique is used as the main smoothing technique throughout this thesis. For other smoothing techniques, the interested reader is referred to (Jurafsky and Martin, 2000) and (Zhai and Lafferty, 2001).

In Jelinek-Mercer smoothing, the maximum likelihood estimates are interpolated with a more general distribution, often called *background model*, or *collection model*; the maximum likelihood estimates are often referred to as *foreground models* or *document models*. The smoothed estimates are calculated as follows:

$$\phi_i = \lambda \phi_{i_{\mathrm{ML}}} + (1-\lambda)\phi_{i_{\mathrm{BG}}}. \tag{3.19}$$

where $\phi_{i_{\mathrm{BG}}} = \mathrm{P}(\mathrm{term}_i)$ is the background probability of observing term_i and λ is a mixing parameter indicating the relative importance of maximum

likelihood estimates. The background probability is usually estimated using either collection frequency, the relative frequency of the term in the collection ($\phi_{i_{\text{BG}}} = \sum_d t_{d,i} / \sum_d \sum_j t_{d,j}$), or document frequency the relative fraction of documents that the term occurs in ($\phi_{i_{\text{BG}}} = df(t_i) / \sum_j df(t_j)$). The mixing parameter λ can be estimated on a training set with known relevant query-document pairs (see Section 4.2.3).

3.6.2 The idf role of smoothing

Besides avoiding the zero-frequency problem, smoothing also serves another purpose, namely that of explaining common query terms and reducing their influence (Zhai and Lafferty, 2001). Because common terms have high background probability, the influence of their foreground probability on the ranking will be relatively small. This becomes apparent when we substitute the ϕs in the retrieval function (Equation 3.7) for the smoothed estimates (Equation 3.19) and do some formula manipulation.

$$\mathrm{P}(\boldsymbol{q}|\boldsymbol{\phi}) = \frac{\left(\sum_{j=1}^{T} q_j\right)!}{\prod_{j=1}^{T} q_j!} \prod_{i=1}^{T} [\lambda\phi_{i_{\text{ML}}} + (1-\lambda)\phi_{i_{\text{BG}}}]^{q_i} \tag{3.20}$$

$$= \frac{\left(\sum_{j=1}^{T} q_j\right)!}{\prod_{j=1}^{T} q_j!} \prod_{i=1}^{T} \left[\frac{\lambda\phi_{i_{\text{ML}}}}{(1-\lambda)\phi_{i_{\text{BG}}}} + 1\right]^{q_i} \prod_{i=1}^{T} [(1-\lambda)\phi_{i_{\text{BG}}}]^{q_i} \tag{3.21}$$

For terms that are not present in the document $\lambda\phi_{i_{\text{ML}}} = 0$ and the corresponding factor reduces to 1. Thus the first product needs only to be considered for query terms that are matched in the document; The latter is document independent and can be ignored for ranking:

$$\mathrm{P}(\boldsymbol{q}|\boldsymbol{\phi}) \propto \frac{\left(\sum_{j=1}^{T} q_j\right)!}{\prod_{j=1}^{T} q_j!} \prod_{i \in \{1,\ldots,T\}: t_i > 0} \left[\frac{\lambda\phi_{i_{\text{ML}}}}{(1-\lambda)\phi_{i_{\text{BG}}}} + 1\right]^{q_i} \tag{3.22}$$

In this last Equation, it is clear that the background probability plays a normalisation role, similar to *idf* in traditional *tf.idf* weighting (Salton and Buckley, 1988). Common terms, i.e., terms with high $\phi_{i_{\text{BG}}}$, contribute less to the final ranking.

3.6.3 Interpolated Gaussian mixture models

The zero-frequency problem does not exist for images, since they are modelled using Gaussian mixture models and Gaussians have infinite support.

However, the second purpose 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 maximum likelihood estimates for the Gaussian mixture models, again interpolation with a more general, background distribution is used. The smoothed version of the likelihood for a single image sample $\boldsymbol{v}$ (cf. Equation 3.5) becomes

$$\mathrm{p}(\boldsymbol{v}|\boldsymbol{\theta}) = \kappa \left[\sum_{i=1}^{C} \mathrm{P}(c_i) \; \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma_i}|}} e^{-\frac{1}{2}(\boldsymbol{v}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1}(\boldsymbol{v}-\boldsymbol{\mu_i})} \right] + (1-\kappa)\mathrm{p}(\boldsymbol{v}), \tag{3.23}$$

where κ is used as the mixing parameter for visual models. A useful way of thinking about this smoothed variant is the following. A sample from a smoothed Gaussian mixture model comes from one of the C components, or from a general background model. The background density $\mathrm{p}(\boldsymbol{v})$ is estimated by marginalisation over all document models in a reference collection $\boldsymbol{\Theta}$:

$$\mathrm{p}(\boldsymbol{v}) = \sum_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathrm{p}(\boldsymbol{v}|\boldsymbol{\theta}) \mathrm{P}(d) \tag{3.24}$$

The reference collection $\boldsymbol{\Theta}$ can be either the current collection, a representative sample, or a separate, comparable collection. The application of interpolation based smoothing to Gaussian mixture models for images is a typical example of the cross-fertilisation between the language models and the image models. Chapter 4 shows that, also in visual retrieval, smoothing with a background collection turns out to be crucial for retrieval performance. *Idf* type weighting for image retrieval is not new (e.g., Squire et al., 1999; Zhu et al., 2002), but interpolation with a more general, background or collection model has not been applied to image retrieval before.

3.6.4 Interpolated language models for video

When a document collection contains video material, we would like to exploit the hierarchical data model of video, in which a video is subdivided in scenes, which are subdivided in shots, which are in turn subdivided in frames. Interpolation based smoothing is particularly well-suited for modelling such representations of the data. To include the different levels of the hierarchy, we can simply extend estimation of the mixture of foreground and background model (Eq. 3.19) with models for shots and scenes:

$$\begin{aligned} \phi_i = \lambda_{\text{Shot}} \mathrm{P}(\text{term}_i|\text{Shot}) + \lambda_{\text{Scene}} \mathrm{P}(\text{term}_i|\text{Scene}) + (\lambda_{\text{Coll}}) \mathrm{P}(\text{term}_i), \\ \text{where} \lambda_{\text{Coll}} = 1 - \lambda_{\text{Shot}} - \lambda_{\text{Scene}} \end{aligned} \tag{3.25}$$

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 model, misalignment between audio and video can be handled. Depending on the information need of the user, a similar strategy might be used to rank scenes or complete videos instead of shots, that is, the best scene might be a scene that contains a shot in which the query terms (co-)occur. Finally, interpolated language models are not only suitable for video retrieval, they can be used in any situation where language has a hierarchical structure. For example, it can be used for passage retrieval from (xml-) documents, where a document can be a hierarchical structure of chapters, sections and paragraphs (Ogilvie and Callan, 2003).

3.7 Generative models, classification and relevance

This chapter introduced generative probabilistic models and explained how they can be used for information retrieval. To summarise, each of the documents in a collection is represented by a probability distribution. These distributions are estimated based on the document's content and smoothed using background probabilities. For retrieval, a query is treated as an observation from one of the models and the models are ranked by decreasing likelihood of generating this observation.

This means information retrieval is treated as a classification problem. In classification problems, the goal is to assign a given observation to one of the pre-defined classes. For example, based on the properties of a blood sample, it is possible to classify the blood type of a person, or to decide whether or not a disease is present in the blood. In image retrieval, a classification view is common: a query is treated as an observation from one of the classes (documents) in the collection and the goal is to find the class from which this observation is most likely drawn (e.g., Vasconcelos, 2000; Fergus et al., 2003; Greenspan et al., 2001; Luo et al., 2003). In the QBE paradigm, such a view is natural: query images and document images are the same type of objects and it seems natural to assume that a query and its relevant document(s) are generated from the same model.

In text retrieval, this approach is less obvious. Queries are of a somewhat different nature than documents. For example, they are typically shorter and often do not consist of full sentences. Indeed there is some controversy around the idea of using language models for information retrieval. The goal of an information retrieval system is to return documents that are relevant to

an information need which is expressed by a query. In language models, the notion of relevance has gone (Sparck Jones et al., 2003) and the goal is to find *the* document that generated the query terms. This also implies there is only *one* relevant document in the collection, an assumption obviously not always true. The absence of relevance hampers the abilities to exploit feedback using language models, since the likelihood that the model of document A generated the query does not change once a user has indicated document B is relevant.

Lafferty and Zhai (2003) argue that the language modelling approach to information retrieval and the classical probabilistic approach to information retrieval (Robertson and Sparck Jones, 1976) are probabilistically equivalent (see also Chapter 5). They are just two ways of decomposing the formulae behind the basic question: *What is the probability that this document is relevant to this query?* When it comes to estimation however, the models are quite different. At some point in the language modelling approach, the assumption has to be made that the probability of observing a query Q conditioned on observing a relevant document D and, is equivalent to the probability that the document generates the query.

Lavrenko and Croft (2003) try to solve the problem of the absence of an explicit notion of relevance in the language modelling approach to information retrieval, by estimating *relevance models* and developing ways to estimate relevance model parameters from small amounts of data like short queries. In Chapter 5, relevance models and relevance feedback will be addressed again. Meanwhile, we simply acknowledge the fact that we assume a document is relevant to a query if the document model is likely to generate the query. Some models are more likely sources then others, but many models can be (somewhat) likely and thus many documents can be relevant to the same query.

4

Experimental results

This chapter focuses on the word *using* in the title of this thesis. How can the models introduced in the previous chapter be used for ad hoc retrieval from generic multimedia collections? Chapter 3 already sketched how generative models can be applied in an information retrieval task (Section 3.4), here the details are filled in and experimental results are reported.

Language models have been evaluated on large-scale text collections before, but never in a multimedia context. Gaussian mixture models have been evaluated on multimedia content, but only on collections that are crafted specifically for a limited retrieval task or on collections that contain clear clusters of images with a high within-cluster similarity and a low across-cluster similarity. Examples of collections for specific tasks are the BRODATZ collection (Brodatz, 1966) for texture retrieval and the COLUMBIA collection (Nene et al., 1996) for retrieval of objects under varying viewpoints. A good example of a clustered collection is the COREL dataset. In the research presented here, both models are tested on generic, heterogeneous multimedia databases.

The chapter begins with a description of the experimental setup in Section 4.1 and tuning the model parameters in Section 4.2. Section 4.3 describes ad hoc retrieval experiments on visual data and Section 4.4 reports experimental results for textual and multimodal data. Finally, Section 4.5 summarises and discusses the main findings.

4.1 Experimental setup

This section describes the setup of the experiments reported in this chapter. All experiments use the basic models introduced in Chapter 3. For each document in a given search collection, a Gaussian mixture model and a language

model are estimated.[1] A query is represented like a document, as a tuple of a textual and a visual document. For retrieval, the approach from Section 3.4 is followed and the sample likelihood of either the visual query document, or the textual query document, or both query documents (depending on the used modalities) is computed to rank the collection.

In this thesis video retrieval is treated as image retrieval and each shot is represented by a *keyframe*, a frame that is representative of the whole shot. For a variant of the models that capture some of the dynamics of a shot see (Ianeva et al., 2004; De Vries et al., 2004b). To find appropriate keyframes, in principal, the Gaussian mixture models can be used to select the frame within a shot that is most similar to all other frames, but this direction is not pursued in this thesis. Instead, naive approaches to selecting representative frames are taken, the details of keyframe selection are discussed as the video test collections are introduced.

The remainder of this section introduces the used test collections (Section 4.1.1) and the features that are used in the models (Section 4.1.2).

4.1.1 Test collections

As explained in Section 2.5.1, a test collection consist of three parts: a set of documents, a set of information needs (or queries) and a set of relevance judgements, indicating which documents are relevant to which queries. In this section, the four test collections used in this chapter are introduced.

Corel3892

The first test collection consist of a selection of COREL images. As already mentioned in Section 2.5.1, one problem with evaluation using COREL is that the data is sold commercially on separate thematic Cd's and a *single* COREL set does not exist. To improve the comparability across different publications, the experiments reported here use the intersection of the 600 classes to which we have access and the classes used by Duygulu et al. (2002) and Jeon et al. (2003). The resulting 39 classes are listed in table 4.9. The 3892 images from these classes form the document set for the first test collection. Each of the documents in the collection is used as a query and the class labels are assumed to represent the ground truth or relevance judgements, i.e, a document is relevant to a query if and only if both are in the same class. This test collection will be referred to as COREL3892.

[1] Background probabilities are estimated on the test collections.

Corel390

A second COREL subset is constructed by randomly taking 10 documents from each of the 39 classes in COREL3892. This smaller set is used for experimenting with many different parameter settings (Section 4.2). Again, all documents in the collection are used as queries and the class labels are used as relevance judgements. This smaller COREL test collection is referred to as COREL390.

Trecvid2002

The third test collection used in this chapter, is the collection used in 2002s TRECVID workshop (Smeaton and Over, 2003). It consists of about 40 hours of video material from the Prelinger archives available from the Internet Archive[2] and the Open Video Project[3]. The set contains advertising, educational, industrial and amateur films produced between 1930 and 1970. The basic unit for retrieval is the shot and the collection comes with a predefined shot segmentation. We reduce video retrieval to image retrieval, by using the middle frame of each shot as the keyframe representative of the whole shot. TRECVID2002 comes with 25 multimedia topic descriptions (information needs) and the corresponding relevance judgements. For an example topic description see the introduction to the TRECVID collections in Section 2.5.1.

Trecvid2003

The TRECVID2003 test set is the collection used at TRECVID a year later (Smeaton et al., 2003a). It consists of 65 hours of ABC and CNN broadcast news from 1998. The collection is shot segmented and comes with a predefined set of keyframes which we use to represent the shots. Again 25 multimedia topic descriptions and the relevance judgements for those are available.

4.1.2 Content representation

Chapter 3 talks about feature vectors that represent a visual sample without specifying any details about the features. This section discusses the features in detail. First the features for the visual models are discussed, then the textual features.

[2] `http://www.archive.org/movies`
[3] `http://www.open-video.org`

Visual features

To be able to retrieve relevant, or visually similar images for a given topic, it is important to have a representation that preserves semantics (Pentland et al., 1996). The image representation for the Gaussian mixture models should capture as much of the information in the images as possible, while being small enough to keep computation tractable. A natural option to consider is image compression algorithms that are designed to yield a concise description of all information in an image. JPEG compression is designed to be state of the art with regard to compression rate and image fidelity and to be not restricted to any particular type of images (Wallace, 1991). This is exactly what is needed here. We want to model generic image collections and to represent image models compactly, while preserving information. JPEG compression is based on the discrete cosine transform (DCT), a cheap approximation of PCA transform, which optimally preserves information. When applied to images, the DCT transform captures both colour and texture information and is thus useful for generic retrieval tasks. We follow the JPEG standard and work in the YCbCr colour space, which separates chromatic information (colour, Cb and Cr channels) and achromatic information (intensity, Y channel).

The process of transforming images to feature vectors is visualised in Figure 4.1. First, an image is converted to the YCbCr colour space, then each channel is cut into blocks of 8 by 8 pixels and the discrete cosine transform (DCT) of the blocks is computed. Then, feature vectors are composed of a fixed number of the most important DCT-coefficients from the Y-channel and a fixed number of the most important coefficients from the Cb and Cr channels. Section 4.2 investigates the optimal number of features from the different channels. From the feature vectors, we build a mixture model using the EM algorithm as described in Section 3.5.1.

To be able to distinguish between images with similar characteristics at different locations, position information is added to the feature vectors, by extending the feature vector with the x and y position of the sample in the image plane. This facilitates differentiating between images with similar features in different locations (e.g., example images with sky at the top versus images with sky all over the image plane). To build the mixture models from feature vectors with position information, two approaches are investigated. In the first approach, position information is treated just as the other features and a Gaussian mixture model is trained in the (DCT,x,y)-domain. The second approach first trains a model on DCT coefficients only and adds the position information after training by calculating the mean and variance of the position information for the samples assigned to each component. In

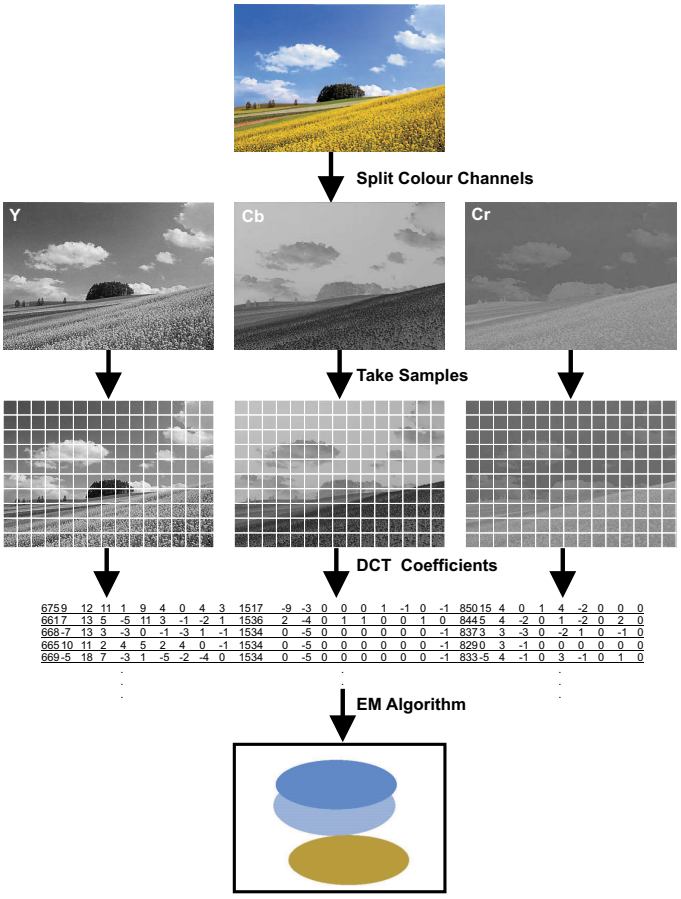


Figure 4.1: Building a Gaussian Mixture Model from an Image.

the second approach, the models are forced to find components with similar colour and texture regardless of the position of this colour and texture. In the first approach the models are restricted to finding components that are coherent in colour, texture and space.

Textual features

The textual features used to estimate language models from, are the words from a text. In this case the texts are automatic speech recognition transcripts for the TRECVID2002 and TRECVID2003 collections and captions and keywords for the COREL3892 and COREL390 dataset. The TRECVID transcripts are produced by LIMSI (Gauvain et al., 2002) and are shot segmented, i.e., a piece of text is associated with each shot. To normalise the word-forms, we stripped suffixes from the words in the transcripts using the Porter stemmer (Porter, 1980) and used a short stoplist to remove some obvious non-content terms. The remaining terms form the vocabulary for indexing and retrieval. Each document is represented as a vector of term counts $\boldsymbol{t} = (t_1, t_2, \ldots, t_T)$, where T is the number of terms in the vocabulary. Term frequencies for shots and scenes are calculated to find the maximum likelihood estimates (cf. Eq. 3.17).

4.2 Tuning the models

This section investigates the effects of using different model parameters on retrieval performance. The main focus is on finding a good set of visual features (Section 4.2.1). Section 4.2.2 discusses the effect of EM initialisation on retrieval results. Finally, Section 4.2.3 investigates the optimal settings for the mixing or smoothing parameters κ and λ.

4.2.1 Varying visual features

The focus of this thesis is more on the models than on the features from which they are build. We are not interested in finding the perfect set of features. However, within the boundaries set by the choice of features as described in Section 4.1.2, some variation is possible. This section tries to find the optimal settings given these boundaries. The number of components in the Gaussian mixture models is varied, as is the set of features that describes a pixel block. The type of features (DCT coefficients and position information) remains fixed. In order to be able to try a large number of different settings in a fair amount of time, a small test collection is needed. Therefore, we

take the COREL390 subset. For each setting, we then build models for this reduced collection. We use each of the 390 collection images as a query and calculate average precision scores for them. The scores are averaged over all queries and we report the mean average precision scores per setting.

In the following experiments the procedure from Section 4.1.2 is used to extract feature vectors, varying the following parameters:

NY: Number of DCT coefficients from Y channel (1, 3, 6, 10, 15 or 21).

NCbCr: Number of DCT coefficients from Cb and Cr channels (0, 1 or NY).

XYpos: Way of using position information of pixel blocks: do not use (denoted by *not*), add to feature vector before training (*pre*), or add to mixture components after training (*post*)

C: Number of Gaussian mixture components (1, 2, 4, 8, 16 or 32).

Testing all possible combinations would yield over 300 different test settings. To reduce this number, first the number of coefficients from the Y channel is fixed. We think both colour and texture are important in image retrieval, thus we need to take a reasonable number of coefficients into account to capture some of the texture information. On the other hand taking too many coefficients does not seem useful; In the JPEG compression literature it is common to ignore the lower coefficients since they do not capture much information anyway. We fix the number of Y coefficients at $NY = 10$. At the end of this section, after having fixed the other parameters, the influence of varying NY is discussed.

The top rows in Figures 4.3–4.5 show visualisations of Gaussian mixture models estimated from the image in Figure 4.2 using different parameter settings; the bottom rows show images constructed by randomly sampling from these models. Models that contain no position information (XYpos=*not*) are visualised by distributing the components' visualisations uniformly across the image plane, in the random images from these the randomly drawn samples are scattered across the image plane.[4]

Table 4.1 shows the results for different values of NCbCr, XYpos and c. The scores do not differ a lot. One could conclude that as long as one uses a mixture (c>1) rather than a single Gaussian (C=1), it does not matter much which model one chooses. However, a small difference in average scores might still be significant: run A might be consistently better than run B, but a few outliers, or errors for which run B is better, can annul this effect and a similar score for both runs can be obtained.

[4]For further explanation of the visualisation of Gaussian mixture models and random samples, see Sections 3.3.2 and 3.4.

Figure 4.2: Example image. Greek Islands.

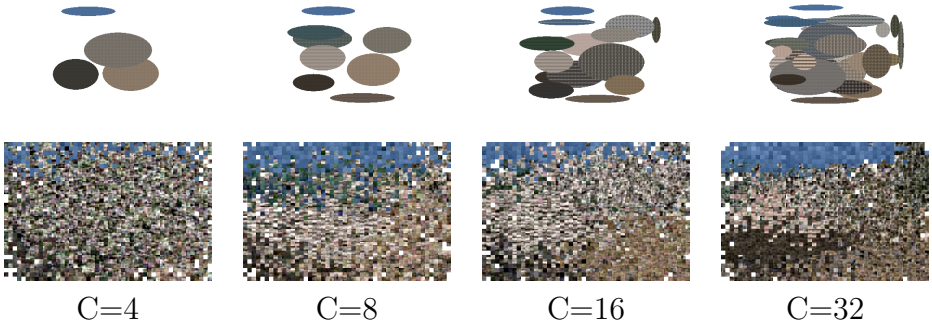


Figure 4.3: Models and random samples. Varying number of components (fixed parameters: NY=10, NCbCr=1, XYpos=*pre*)

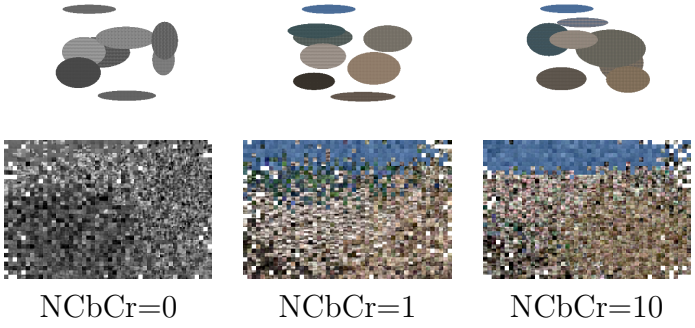


Figure 4.4: Models and random samples. Varying number of colour coefficients (fixed parameters: NY=10, XYpos=*pre*, C=8)

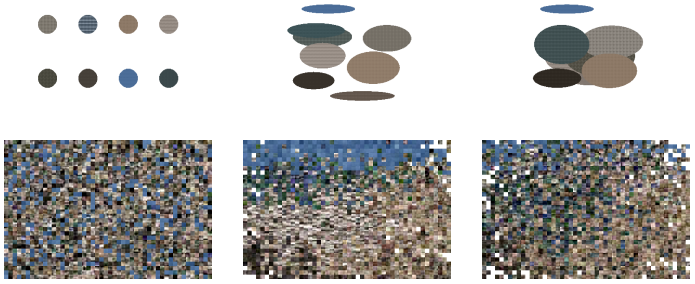


Figure 4.5: Models and random samples. Varying position information, (fixed parameters: NY=10, NCbCr=1, C=8) For visualisation purposes, the components in the XYpos=*not* setting are distributed uniformly across the image plane, while in fact no position information is available in this setting.

Table 4.1: mean average precision scores for different parameter settings (fixed: NY=10).

NY	NCbCr	XYpos	C=1	C=2	C=4	C=8	C=16	C=32
10	0	*not*	0.08	0.18	0.20	0.21	0.21	0.21
10	0	*pre*	0.09	0.19	0.21	0.21	0.21	0.20
10	0	*post*	0.09	0.19	0.21	0.21	0.22	0.21
10	1	*not*	0.13	0.22	0.23	0.23	0.23	0.23
10	1	*pre*	0.13	0.22	0.23	0.23	0.23	0.22
10	1	*post*	0.13	0.22	0.23	0.24	0.23	0.23
10	10	*not*	0.12	0.22	0.23	0.24	0.24	0.23
10	10	*pre*	0.13	0.21	0.24	0.24	0.24	0.23
10	10	*post*	0.13	0.22	0.23	0.24	0.24	0.23

Table 4.2: Top scoring parameter settings based on pair-wise comparisons.

NY	NCbCr	XYpos	c	#better	#equal	#worse
10	10	*pre*	8	0	8	45
10	10	*post*	8	0	9	44
10	10	*not*	8	0	11	42
10	10	*not*	16	0	12	41
10	10	*pre*	16	0	12	41
10	10	*post*	32	0	20	33
10	10	*pre*	4	0	23	30
10	10	*not*	32	1	19	33
10	1	*post*	8	1	19	33
10	10	*pre*	32	3	18	32

To test whether the differences between the various settings are statistically significant, the paired Wilcoxon signed-rank test[5] is used (significance level of 5%), for each pair of parameter settings. Since we report on many independent statistical tests, each performed at a significance level of 5%, we can expect to make some mistakes, i.e., decide there is a significance difference when there is not, or miss a significance difference that is there. Therefore, we will not attribute much value to individual outcomes of the significance tests. Instead, we mainly look at trends when varying a single parameter.

For each setting we count the number of other parameter settings that are significantly better, the number of significantly worse models and the number of other models that do not differ significantly. Table 4.2 shows the best models according to these counts (i.e., the models with the lowest number of other models that are significantly better). While the top scoring models share a few interesting properties (like the number of CbCr coefficients used), it is more informative to look at the influence of changes in the individual parameters. In the remainder of this section, we consecutively discuss the influence of changing C, NCbCr and XYpos.

When varying the number of components in the mixture model, it can be expected that a low number gives insufficient resolution to describe all image samples well, whereas a high number of components is bound to result in over-fitting. Table 4.3 shows the experiments confirm this intuition. Using more components initially gives significant improvements, but at C=8, we

[5]The Wilcoxon test is introduced in Section 2.5.3

Table 4.3: Comparing different models, varying the number of components c (with fixed NCbCr=10, XYpos=*post*). The numbers indicate if we see a significant difference when changing from row to column setting: 0 for no significant difference; 1 for significant improvement; -1 for significant deterioration.

c	1	2	4	8	16	32
1	0	1	1	1	1	1
2	-1	0	1	1	1	1
4	-1	-1	0	1	1	0
8	-1	-1	-1	0	0	0
16	-1	-1	-1	0	0	-1
32	-1	-1	0	0	1	0

reach an optimum. After that no significant improvements are measured and sometimes using more than 8 components even harms results (probably because of over-fitting). Comparable results are found for settings of NCbCr and XYpos which are not shown in Table 4.3. For some settings the optimum is already reached at C=4.

Colour information is generally assumed to be a valuable source for the purpose of image retrieval. When we vary the number of coefficients used from the Cb and Cr channels (NCbCr), we see that colour information is important indeed. Both NCbCr=1 and NCbCr=10 yield significantly better scores than NCbCr=0 for each setting of XYpos and c. This shows it is important to use at least 1 DCT coefficient from each colour channel and thus to encode colour information in the models. For some settings in which more components (c$\geq$8) are used, using 10 coefficients from the colour channels is significantly better than using only 1 (see Table 4.4 for an example). So, it seems wise to use as much colour information as possible for describing the images, as long as the models can accommodate all this information, i.e., as long as there are enough components. With fewer components, using only 1 coefficient is better.

Finally, it is unclear how varying the use of position information (XYpos) influences the scores. For many settings of c and NCbCr, there is no significant difference between different settings of XYpos. When there is, it is sometimes an improvement, sometimes a deterioration. Only in the single component case (C=1), there is a consistent significant improvement for models that do use position information. However, when a single component is used to describe an image, all samples must be assigned to that one component and the position of this single component must be the centre

Table 4.4: Comparing different models, varying the number of DCT coefficients from the colour channels NCbCr (with fixed XYpos=*pre*, C=8). See Table 4.3 for explanation.

NCbCr	0	1	10
0	0	1	1
1	-1	0	1
10	-1	-1	0

of the image plane with a variance related to the size of the image. However, the position information is different for portrait and landscape images since the position of the blocks in these will have different variance. Thus, adding position information in the single component case, acts as a portrait vs. landscape classifier; apparently this improves retrieval results. This is not surprising considering the fact that images within a single class tend to have the same orientation.[6] One thing that *can* be learnt from analysing the results for different XYpos settings is that it never harms to use position information: in all cases, using it either significantly improves results or it does not change results. Still, the experimental results do not clarify whether we should incorporate this information directly (XYpos=*pre*) or after training the models (XYpos=*post*).

The optimal settings found so far are NCbCr$\geq$1, XYpos $\in\{$*pre*,*post*$\}$ and C=8. In the following experiments, these values are fixed and different values for NY are evaluated. The experiments incorporate position information before training (XYpos=*pre*) and use either 1 DCT coefficient from the colour channels (NCbCr=1), or as many as from the intensity channel (NCbCr=NY). Since the dimensionality of the features space gets higher as NY increases, it can be expected that more components are needed to capture the information. Therefore, experiments with both 8 (the optimal so far) and 16 components are carried out. The results are listed in Table 4.5. Significance test results, for C=8 are shown in Tables 4.6 and 4.7 (C=16 yields comparable results). For NCbCr=1, using more coefficients from the Y channel is significantly better Apparently, when the colour channels cannot capture texture information, adding more texture from the intensity (Y) channel helps. When texture from the colour channels is taken into account as well (NCbCr-NY), adding more texture from the Y-channel does not help. It seems using a fair amount of texture information is useful. However, to keep computation tractable, we fix the feature and model parameters at the

[6]For 75% of the classes, over 70% of the images has the prevalent orientation.

Table 4.5: Mean average precision for different values of NY; NCbCr∈{1,NY},C∈{8,16}, XYpos=*pre*

	C=8		C=16	
NY	NCbCr=1	NCbCr=NY	NCbCr=1	NCbCr=NY
1	0.20	0.20	0.20	0.20
3	0.21	0.22	0.21	0.22
6	0.22	0.24	0.22	0.23
10	0.23	0.24	0.23	0.24
15	0.24	0.24	0.24	0.24
21	0.24	0.24	0.24	0.24

Table 4.6: significance test for varying NY; XYpos=*pre*, C=8, NCbCr=1. See Table 4.3 for explanation.

NY	1	3	6	10	15	21
1	0	1	1	1	1	1
3	-1	0	1	1	1	1
6	-1	-1	0	1	1	1
10	-1	-1	-1	0	1	1
15	-1	-1	-1	-1	0	1
21	-1	-1	-1	-1	-1	0

following (sub-optimal) setting: NY=10, NCbCr=1, XYpos=*post*, C=8.

4.2.2 EM initialisation

It is a well known fact that the EM algorithm is sensitive to its initialisation. Building the Gaussian mixture models starts from a random initialisation, thus we may end up with different models if we build two models from the same frame. This Section investigates the influence of the EM initialisation on the final ranking of the documents. To do so, we build a collection with several models for each frame and compare the scores of the different versions on a number of queries. We concentrate on the effects of initialisation on top ranking documents, i.e., the documents that are most similar to the query. For this purpose, a subset from TRECVID2002 is taken to get a collection with different levels of (assumed) similarity. The collection is produced using the following procedure.

- Select two videos from the TRECVID2002 collection;

Table 4.7: significance test varying NY; XYpos=*pre*, C=8, NCbCr=NY

NY	1	3	6	10	15	21
1	0	1	1	1	1	1
3	-1	0	1	1	1	1
6	-1	-1	0	0	1	0
10	-1	-1	0	0	0	0
15	-1	-1	-1	0	0	0
21	-1	-1	0	0	0	0

- From each shot in each of the videos select five frames (evenly distributed over the shot);
- For each frame build 10 models (using EM from random initialisations).

This way, it is possible to differentiate between exact matches (i.e., different models of the same frame), frames from the same shot, frames from the same video and frames from different videos. The middle frames of each shot are used as a queries.

For a given query, different models from the same frame are expected to have roughly the same scores. In addition, the scores for models of other frames from the same shot should not vary much either.

Scores are calculated as follows. The collection is ranked for a given query. For each frame in the collection, the average rank of all 10 models representing the frame is computed, as well as the standard deviation from this average. These scores are then averaged over all frames in a given set and over all queries. The sets of frames considered are: the single frame, all frames from a shot, all frames from the same video and all frames in the collection. The results are shown in table 4.8.

If EM was insensitive to its initialisation, all models for a given frame would have been exactly the same and they would have been ranked in sequence, yielding the best possible standard deviation for 10 models: 3.03. The table shows that this is not the case, thus indeed, initialisation influences EM. However, on average all 10 models of the query frame are near the top of the ranked list (mean rank 8.06 std dev. 5.95). Furthermore, different models of frames from the same shot are on average closer together (and closer to the top of the list) than models from other frames. In fact, the mean rank of a set of frames correlates with the standard deviation of this rank. Frames that rank higher are in general closer together. Figure 4.6 shows mean rank and standard deviation for different queries in a single plot. On the x-axis the different queries are listed, sorted by mean shot-rank (i.e.,

Table 4.8: Mean rank with standard deviation for different models of the same frame. Averaged over all queries and over different sets of frames

	ranks	
set	mean	std-dev
frame	8.06	5.95
shot	269.85	35.09
video	2946.10	286.15
collection	3075.50	374.02

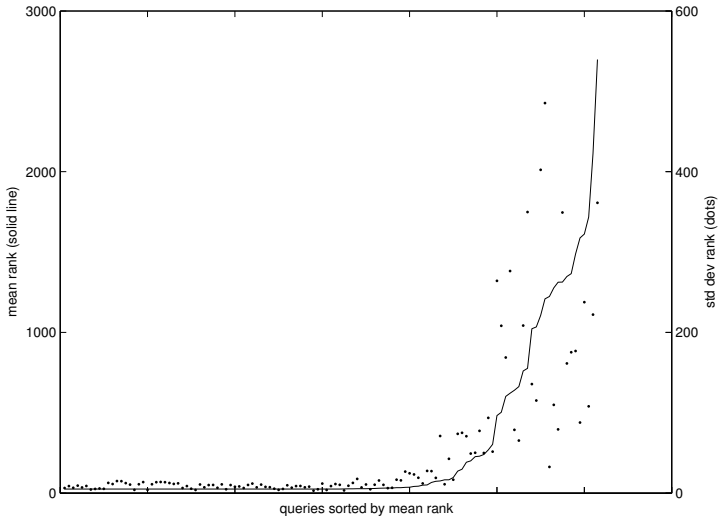


Figure 4.6: Mean and standard deviation of shot ranks for different queries

the mean rank of all models of all 5 frames from the same shot as the query). The solid blue line corresponds to the left y-axis and shows the mean shot-ranks for each query; the green dots correspond to the right y-axis and show the standard deviation in shot-rank (i.e., the standard deviation in ranks within this set of 10 models of 5 frames from the same shot as the query frame). The plot shows that if the queries get harder (i.e., mean ranks for frames from same shot get higher), the different models of frames from the query shot get more spread out (i.e., standard deviation goes up). We can conclude that although EM is sensitive to its initialisation, this mainly has an effect on the lower part of the ranking. The top ranking documents are less sensitive to differences in the EM starting points.

4.2.3 Estimating mixing parameters

The smoothed estimates for the generative models are computed as a mixture of foreground and background probabilities (see Section 3.6). For textual video retrieval, this has been extended to a mixture of shot, scene and background probabilities. In both the textual and visual case, the final probability is a weighted sum of the probabilities given particular models. In this section, optimal values for the weights, or mixing parameters, are estimated. To do so, we take a collection with known relevant query-document pairs and evaluate retrieval effectiveness for different values of the mixing parameters.

Visual: κ

Figure 4.7 shows the mean average precision scores on the TRECVID2002 collection for κ ranging from 0.0 to 1.0. It is apparent that retrieval results are insensitive to the value of the mixing parameter, as long as we take both foreground and background into account (i.e., $\kappa > 0$ and $\kappa < 1$). This means smoothing is important for the success of the model; an unsmoothed version ($\kappa = 1$) performs poorly. The plot has a similar shape as that found in Hiemstra's thesis for the λ parameter in the standard language model (Hiemstra, 2001). The optimal value though, $\kappa = 0.90$, differs significantly from the typical language model optimal $\lambda = 0.15$. A possible explanation is the fact that Gaussian mixture models are already smooth distributions. The only reason to introduce Jelinek-Mercer smoothing is to decrease the influence of common samples in a query. In text retrieval, smoothing also has to take care of the zero-frequency problem. In the remainder of this chapter the visual smoothing parameter is fixed at $\kappa = 0.90$.

Textual: λ

The hierarchical video model interpolates shot, scene and background probabilities (see Section 3.6.4). Preliminary experiments with 2001s TRECVID collection gave the parameter setting used throughout this thesis: $\lambda_{\text{Shot}} = 0.090$, $\lambda_{\text{Scene}} = 0.210$ and $\lambda_{\text{Coll}} = 0.700$. This section investigates whether these are the optimal settings by exhaustively searching the parameter space using the TRECVID2002 collection. Since no scene segmentation is available for this collection, we simply assume a scene is a window of 5 consecutive shots. Figure 4.8 shows a surface plot of the mean average precision for different combinations of λ_{Shot} and λ_{Scene}, in all combinations, $\lambda_{\text{Coll}} = 1 - \lambda_{\text{Shot}} - \lambda_{\text{Scene}}$. The plot shows, it is important to incorporate scene information ($\lambda_{\text{Scene}} = 0$ leads to low mean average precision), while ignoring shot information seems to be less influential. In fact, when $\lambda_{\text{Shot}} = 0$,

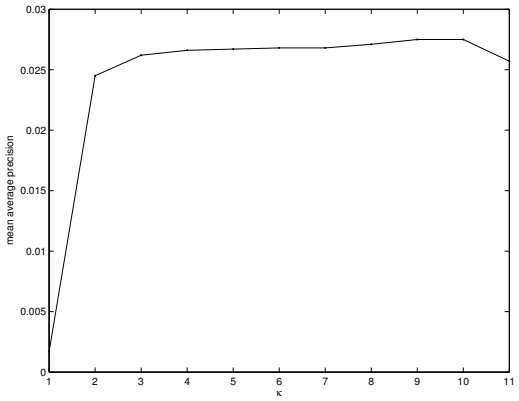


Figure 4.7: Mean average precision on TRECVID2002 search task for different values of κ

the optimal value for the other λs corresponds to optimal values often used in text retrieval: $\lambda_{\text{Scene}} = 0.150$ and $\lambda_{\text{Coll}} = 0.850$. The optimal values for the mixing parameters in the hierarchical language model are reached in the area with both λ_{Shot} and λ_{Scene} close to zero, i.e., the area with high λ_{Coll} (this is most clearly visible in the contour plot underneath the surface). The optimal values for the mixing parameters in the hierarchical language model are reached when all levels of the hierarchy contribute: $\lambda_{\text{Shot}} = 0.010$, $\lambda_{\text{Scene}} = 0.040$ and $\lambda_{\text{Coll}} = 0.950$. The surface around this optimum is rather flat, therefore, any parameter setting within this area is close to optimal. The setting found with 2001s TRECVID collection and used throughout this thesis, lies in this area.

4.3 Visual search

With the optimal parameters found in the previous section, more experiments are conducted. This section focuses on visual information retrieval using the query-by-example paradigm. The textual part of documents is ignored. Throughout the section, the basic models from Chapter 3 are used. Gaussian mixture models are built for all documents in a collection and the sample likelihood of the query is computed to rank documents. In the following subsections, the representation of the query is varied. First, to set a baseline, all available visual examples are used (Section 4.3.1). Then in Section 4.3.2 a query is constructed from manually selected good, or highly representative examples. Finally Section 4.3.3 investigates how selecting important regions in visual examples influences the retrieval results.

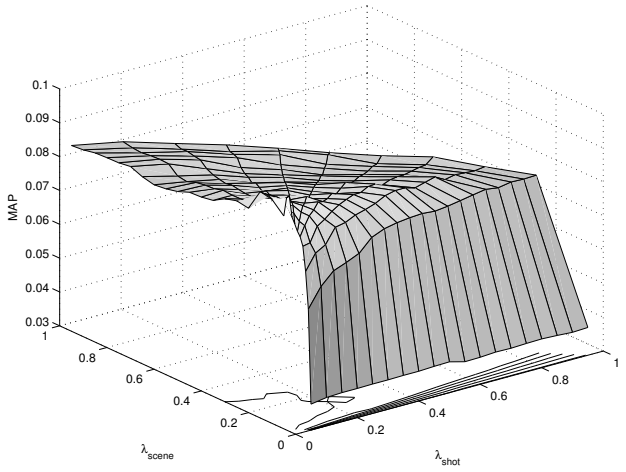


Figure 4.8: Mean average precision on TRECVID2002 for different λs.

4.3.1 All examples

In the most basic variant of the retrieval models, all available visual examples are regarded as a single bag of samples and the sample likelihood of the whole bag is computed. This means, the goal is to find document models which explain, or can generate, all samples in all example images. This comes down to an AND type of query: we want to find documents that explain example A and example B and example C.... This strategy is evaluated on COREL3892, TRECVID2002 and TRECVID2003.

Corel3892

We run each document in COREL3892 as a query[7], rank the full document set (i.e., we do not take a top K) and compute average precision values for each query. We then compute mean average precision per image class. Since there is some variety in the specificity of the classes, some classes might be harder than others (something as specific as *English pub signs* might be more easy than a generic class like *Israel*). Table 4.9 shows the scores for the individual classes (sorted from high to low). Indeed, we see a fair amount of variation (.05 to .36). Figure 4.9 shows an example of a query with the top 5 documents from one of the classes with the highest scores: *Arabian Horses*.

To get more insight into what is actually retrieved for a given query, we can look at confusion between classes. For example, to see how often we confuse *lions* for *tigers*, we can use a image from the *tigers* theme as a query

[7]Note that, in contrast to the experiments described in the previous section, all 3892 images from the 39 classes are used.

Table 4.9: Mean average precision per class for COREL3892 data set.

Class	MAP	Class	MAP
English Pub Signs	.36	Israel	.09
English Country Gardens	.33	Beaches	.09
Arabian Horses	.31	Holland	.08
Dawn & Dusk	.21	Hong Kong	.08
Tropical Plants	.19	Sweden	.07
Land of the Pyramids	.19	Ireland	.07
Canadian Rockies	.18	Wildlife of the Galapagos	.07
Lost Tribes	.17	Hawaii	.07
Elephants	.17	Rural France	.07
Tigers	.16	Zimbabwe	.07
Tropical Sea Life	.16	Images of Death Valley	.07
Exotic Tropical Flowers	.16	Nepal	.07
Lions	.15	Foxes & Coyotes	.06
Indigenous People	.15	North American Deer	.06
Nesting Birds	.13	California Coasts	.06
Images of Thailand	.13	North American Wildlife	.06
Greek Isles	.10	Peru	.05
Cowboys	.10	Alaskan Wildlife	.05
Mayan and Aztec Ruins	.09	Namibia	.05
Wildlife of Antarctica	.09		
	mean	.12	

and look at the number of *lions* we retrieve. This can be done by assuming a retrieved image is relevant to the tiger query if and only if it is from the lions class. Based on these cross-category assessments mean average precision can be computed like before. This process is repeated for all pairs of classes in the collection and the resulting confusion scores are shown in Figure 4.10. Along the y-axis, the query class is plotted, the x-axis shows the class that was assumed relevant to the query. Thus looking at *row X* we can learn what we find if we search for X and *column X* shows what would be a good query to retrieve X.

The diagonal of the figure is darker, indicating that, on average, queries retrieve more images from their own class than images from a different class. Some interesting confusions are found: When querying for *Beaches* we also find *Greek Islands*, a query for *Tropical Plants* returns also *Tropical Sea life* and searching for *Indigenous People* we find *Lost Tribes*. Moreover,

Query:

Results:

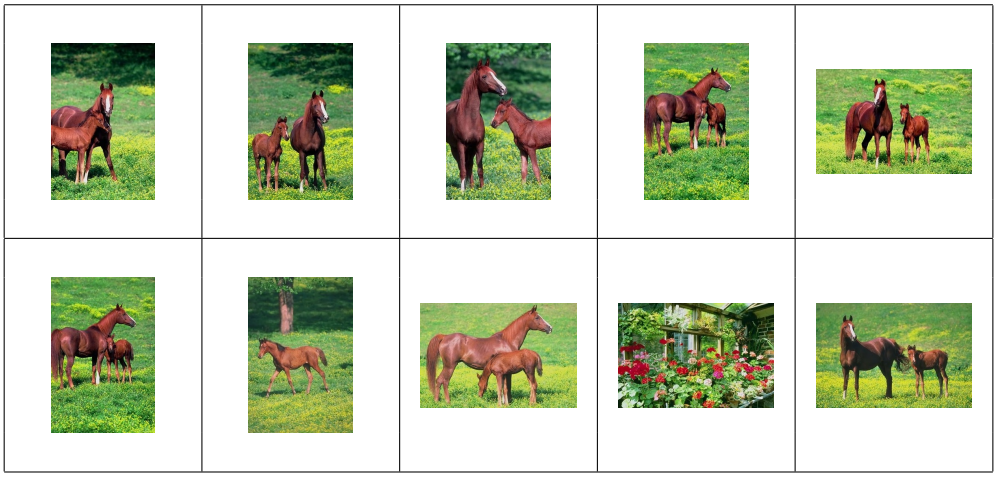


Figure 4.9: Example COREL390 query with top 10 documents.

we see some lighter and darker columns showing that some classes get retrieved hardly ever when using examples from outside that class (*Wildlife of Antarctica*, *Dawn & Dusk*) and others are returned more often for any query (*Indigenous People*, *Lost Tribes*). Also noticeable is the fact that country gardens and tropical plants get mixed up sometimes and that both these classes are retrieved relatively often when Arabian horses are used as query examples. The latter is probably due to the similarity in background: all have green, grassy backgrounds.

Trecvid

The experiments on TRECVID2002 and TRECVID2003 use for each topic a single query composed of all available image examples as well as the keyframes for the example videos.[8] Results are evaluated on the top 100 retrieved shots for TRECVID2002 and on the top 1000 for TRECVID2003.[9] Table 4.10 lists the results in the columns labelled *all* (results in the other

[8]The collections are described in Sections 4.1.1 and 2.5.1.

[9]Conform the requirements from the respective guidelines.

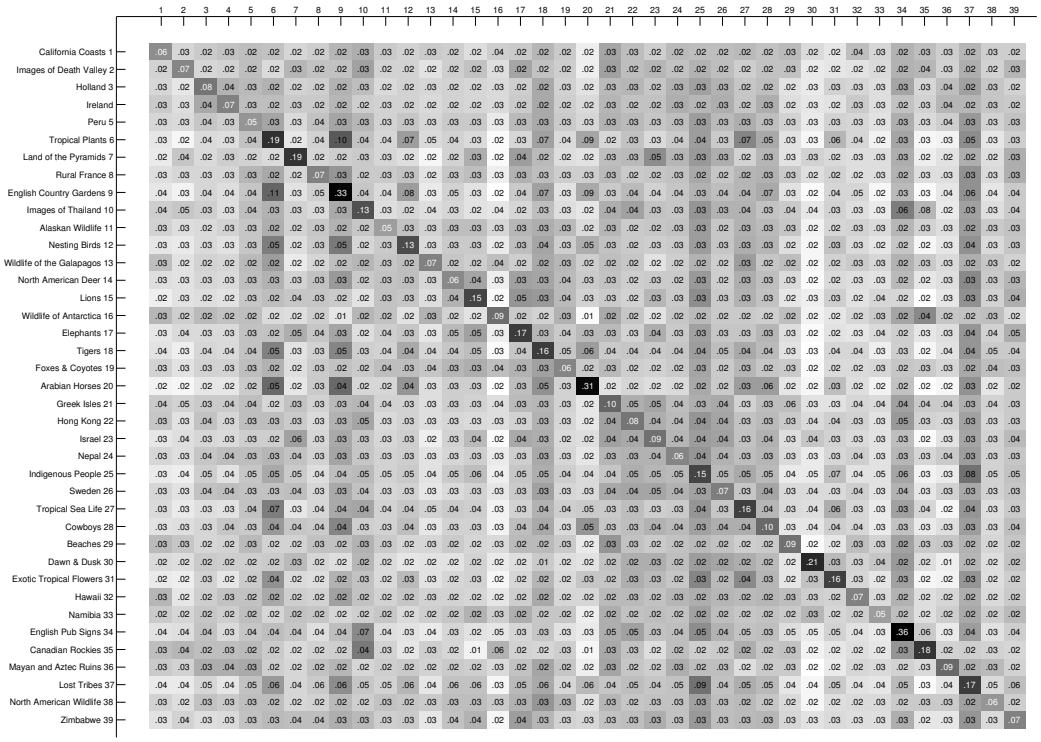


Figure 4.10: Confusion between classes. Mean average precision for different classes of relevant images. Darker squares indicate higher scores. Vertically, the query classes are listed, horizontally, the class that is assumed to be relevant

columns are discussed in the next section). The first thing that can be noted from these results is that the scores are much lower than for the COREL3892 collection. Apparently TRECVID2002 and TRECVID2003 are much harder tasks (at least for the Gaussian mixture models). For some topics the results are reasonable, but most have low scores. Looking in more detail at the results for the better scoring topics, we see that often these are almost known-item topics. The correct results for these are (near) exact matches of the query example(s). For some TRECVID2002 topics (e.g., VT0076: Find shots of Eddy Rickenbacker), some of the topic examples are taken from the search collection. Given the nature of our models, it is not surprising that we are able to retrieve these shots. In other cases, highly specific shots are asked for, like *Dow Jones graphics showing a rise* (VT0120). The Dow Jones graphics in the collection all look highly similar and can thus be found relatively easy. Whether they show a rise or fall of the index is probably not captured in the model, but in the retrieved results there is bound to be a number of rises, yielding a relatively high score for this topic. It remains questionable though how useful such a result is in an actual retrieval setting. Still, for other topics, the models seem useful. For example VT0102, *Find shots from behind the pitcher in a baseball game as he throws a ball that the batter swings at*, retrieves good results. Apparently the document models capture the main characteristics (grass, crowd) of such shots.

It is somewhat difficult to compare these results to those of other groups participating in TRECVID, since there are few groups that submitted a content-based image retrieval run based only on the available visual examples. Many runs make use of the speech transcripts that are available. In addition, many groups use a high level classification of the content and nature of the shots (e.g., indoor, outdoor, people, face, animal). A user then has to formulate queries in terms of these high level classes. For instance, for the baseball query (VT0102) one could ask for *outdoor scene*, *sporting event* and *people*. Developing such high level classes requires a lot of effort. Large amounts of training data need to be carefully selected and manually annotated. Furthermore, the number of classes is necessarily finite and the nature of the classes restricts the type of queries that can be answered. Consider for example how one should formulate a request for shots of a mug or cup of coffee (VT0121) in terms of high level classes, when no class for mugs or cups is available. The rationale behind the models developed and tested in this thesis is that they should be able to answer generic queries, without relying on pre-defined sets of concepts that can be recognised.

Although our approach differs considerably from most of the other research groups participating in TRECVID, it is useful to report scores for other systems and to see where Gaussian mixture models are positioned.

Table 4.10: Average precision per topic for all, best and designated example runs

TRECVID2002			
topic	all	best	des
vt0075	.0038	.2438	.2438
vt0076	.4854	.4323	.1760
vt0077	.0000	.0000	.0000
vt0078	.0000	.0000	.0000
vt0079	.0000	.0040	.0000
vt0080	.0048	.0977	.0977
vt0081	.0000	.0000	.0000
vt0082	.0330	.0234	.0234
vt0083	.0000	.0000	.0000
vt0084	.0046	.0046	.0046
vt0085	.0000	.0000	.0000
vt0086	.0053	.0704	.0704
vt0087	.0000	.0000	.0000
vt0088	.0046	.0069	.0069
vt0089	.0000	.0000	.0000
vt0090	.0000	.0305	.0305
vt0091	.0095	.0095	.0095
vt0092	.0003	.0106	.0000
vt0093	.0006	.0006	.0000
vt0094	.0021	.0021	.0021
vt0095	.0000	.0000	.0000
vt0096	.0323	.0323	.0323
vt0097	.1312	.1408	.0000
vt0098	.0000	.0003	.0003
vt0099	.0000	.0000	.0000
MAP	.0287	.0444	.0279

TRECVID2003			
topic	all	best	des
vt0100	.0004	.0065	.0030
vt0101	.0511	.1105	.0003
vt0102	.3043	.4302	.3529
vt0103	.0000	.0000	.0000
vt0104	.0000	.0010	.0000
vt0105	.0003	.0021	.0021
vt0106	.0000	.0002	.0000
vt0107	.0001	.0163	.0004
vt0108	.0001	.0004	.0003
vt0109	.0002	.0014	.0000
vt0110	.0003	.0011	.0000
vt0111	.0000	.0004	.0004
vt0112	.0001	.0043	.0009
vt0113	.0000	.0027	.0006
vt0114	.0007	.0007	.0006
vt0115	.0004	.0200	.0200
vt0116	.0000	.0000	.0000
vt0117	.0286	.0337	.0112
vt0118	.0000	.0000	.0000
vt0119	.0000	.0000	.0000
vt0120	.3154	.2187	.2187
vt0121	.0003	.0295	.0003
vt0122	.0002	.0048	.0010
vt0123	.0001	.0006	.0006
vt0124	.0000	.1209	.0000
MAP	.0281	.0402	.0245

For TRECVID2002 the mean average precision across all submitted runs in the manual task[10] is .056. If we ignore runs that make use of the speech transcripts, this drops to .044. For TRECVID2003 these numbers are .085 and .021 respectively. The Gaussian mixture model results are just below these averages for TRECVID2002 and just above for TRECVID2003 (see Table 4.10). Considering the fact that the Gaussian mixture models require little a priori knowledge and no manual effort in developing training data, we can conclude that these are good results.

4.3.2 Selecting and combining examples

Treating multiple visual examples as a single bag of examples to retrieve a set (or ranked list) of similar documents can be problematic. Consider for example the topic shown in Figure 4.11. 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 retrieved shot should be similar to any of the examples, but not necessarily to all.

Figure 4.11: VT0101: 'Find shots of a basket being made'.

A common approach to handling multiple example queries is to run separate queries for each example. The final score for a document could then be based on the results for a single (manually selected) representative example, or it could be a function of the results for the individual examples. In this section both approaches are investigated.

Single example queries

A posteriori, it is possible to compute which of the available visual examples gave the best results. A priori, one can only try to guess what would be a good example. For TRECVID2002 and TRECVID2003, we a priori select examples to be used in single example runs, i.e., the ones expected to be good.

[10]In the manual task, there is no user feedback involved. A user has one go at transforming the topic into a query and that query is run through the search system.

We call these *designated examples*. Then we compute the likelihood for each available example separately and compare runs with designated examples (a priori selected), best examples (a posteriori) and all examples. Table 4.10 shows the results. The best performance is obtained when the best example is selected a posteriori. Of course, this is 'cheating', but it shows that indeed combining multiple visual examples in a single query can degrade results. However, manually selecting good single examples for each topic turns out to be difficult, simply using all available examples gives better results than using the designated ones.

Combining examples

A second common approach to dealing with multiple examples in a query 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 (e.g., Jin and French, 2003; Westerveld et al., 2003a; Natsev and Smith, 2003). In fact the approach described in Section 4.3.1, where we treated multiple examples as a single set of samples and computed the joint likelihood of all of them, can be viewed as an example of a simple score combining strategy. The final score for the whole set of examples is the product of the individual example probabilities (*find documents that explain and A and B and C*). In general, however, it is far from trivial to choose a combination function that works well for a variety of queries. This section only looks at a basic rank-based combination technique. Other techniques for handling multiple example queries are discussed in Section 5.2. The rank-based technique discussed here is a simple approach which combines the results from the individual example runs in a round-robin fashion: the final ranked list is composed of all ranks 1 from the individual topics, followed by all ranks 2, followed by all ranks 3, etc. (duplicate results are filtered out). The round-robin combination is evaluated on TRECVID2003, yielding a mean average precision of .0319, slightly higher than the joint likelihood baseline (.0281). Even though it is not possible to manually pick a single best example, simply treating all examples as one large set of samples is not the best strategy. As the basketball example in the introduction to this section showed, some kind of OR-functionality is needed. The round-robin strategy provides this. One could expect the final result list for the basketball example to contain a mixture of good matches for the close-up shots and good matches for the overview shots. The joint likelihood from the baseline implements an AND-strategy, the best models are those that explain both the close-ups and the overviews. The results for the round-robin approach, an OR-strategy, are indeed better for multiple

Figure 4.12: Selected region from a visual example for VT0105: Find shot of a helicopter in flight or on the ground.

examples queries.

4.3.3 Selecting important regions

In the previous sections, query examples are presented to the retrieval system without modification. However, it may be useful to zoom in on specific parts of a query (Baan et al., 2001; Bosch et al., 2001). If a user indicates what is most important in an example image, or why something is a good example, this could help in getting queries more focused. An obvious way to indicate important parts in a query image is to select significant regions. For example for a query for shots of a helicopter in flight or on the ground (TRECVID2003, VT0105), one could draw a bounding box around the helicopter to indicate that the sky in the background is not important (See Figure 4.12). An alternative is to exploit the fact that each component of the Gaussian mixture model describes a different part of the image. The idea is to manually select a subset of the components in the following way. The model defines a soft clustering of the samples in the image, i.e., a sample $\boldsymbol{v_j}$ is assigned to component c_i with probability $\mathrm{P}(c_i|\boldsymbol{v_j})$ (cf. Equation 3.12). This can be turned into a hard clustering by assigning each sample to the most likely component c_k ($k = \arg\max_i \mathrm{P}(c_i|\boldsymbol{v_j})$). The sample assignments can then be presented to a user as a colour coded version of the image, where each colour represents a set of samples assigned to one of the components. A user can then select those clusters that are most meaningful for the query at hand. Figure 4.13 shows an example.

Both region selection and component selection approaches are tested using TRECVID2003. Regions and components are manually selected for each of the examples. The samples from the selected parts (regions or components), are then used as before: the maximum likelihood for the selected subset of samples is computed and used to rank the documents in the col-

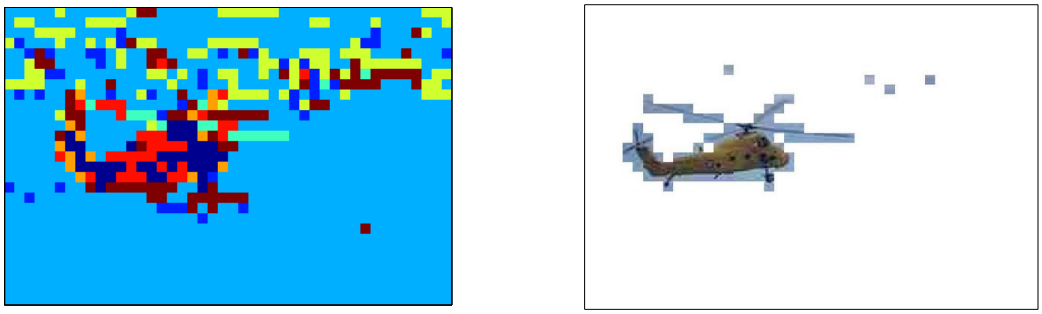


Figure 4.13: Colour coding of sample assignment (left) and selected components (right) for a visual example for VT0105: Find shot of a helicopter in flight or on the ground.

Table 4.11: Selecting Regions or components from TRECVID2003 examples

Query Type	Full Image	Selected Region	Selected Component(s)
AllEx	.0281	.0264	.0285
DesEx	.0245	.0217	.0284

lection. Again a distinction is made between using all examples and using designated examples only. The results are shown in Table 4.11. Selecting regions gives a slight decrease in score, selecting components yields a small improvement.

An explanation for this difference can be found in the homogeneity of the selected parts. Selected components are typically more homogeneous than selected regions. A manual inspection of the retrieval results for the homogeneous component queries shows that results are often intuitive, i.e., there is a clear visual similarity between the query blocks and the top retrieved documents. If the query samples also have a clear semantics (e.g., *sky*), then the results are often useful (Figure 4.14). Sometimes however, a component carries no true semantics. In these cases, results are merely visually similar. Figure 4.15 shows examples of this: looking at the components without the context of the full example, the audience can no longer be identified as such and the grass looks like water. Consequently, the results are still visually similar, but no longer meaningful.

Selecting components can harm results in the case of near exact matching. To illustrate this, the horses query from the COREL390 dataset (Figure 4.9) is revisited. Searching with only manually selected relevant components from this example (i.e., the components that compose the horse), retrieves other brown things, but few horses (See Figure 4.16). However, when we inverse

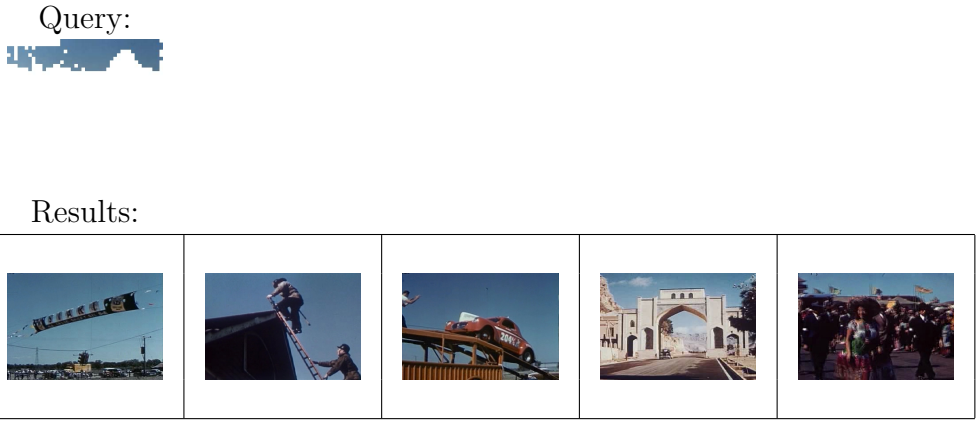


Figure 4.14: Top 5 results for a homogeneous query with clear semantics ('Sky')

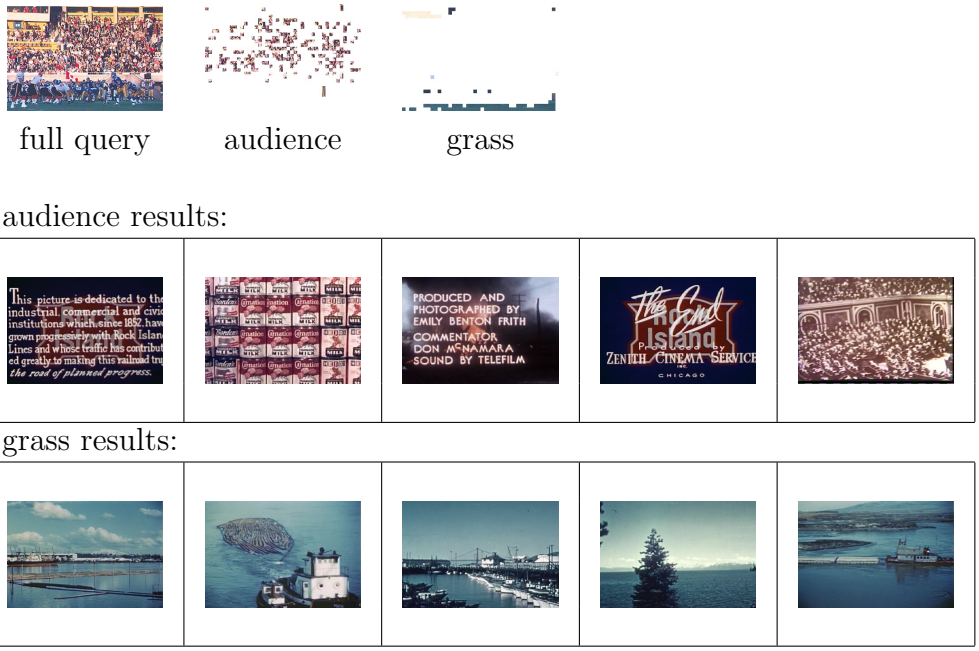


Figure 4.15: Top 5 results for homogeneous queries without clear semantics

the selection and search with only the grass components, the horses are found (Figure 4.17). Clearly the horses we find are near exact matches; all photos are shot in the same field under the same conditions. In a larger, more heterogeneous dataset, more horses without grass and vice versa can be expected, resulting in more realistic scores. The problem will be revisited in Chapter 6, where the characteristics of test collections are discussed.

Query:

Results:

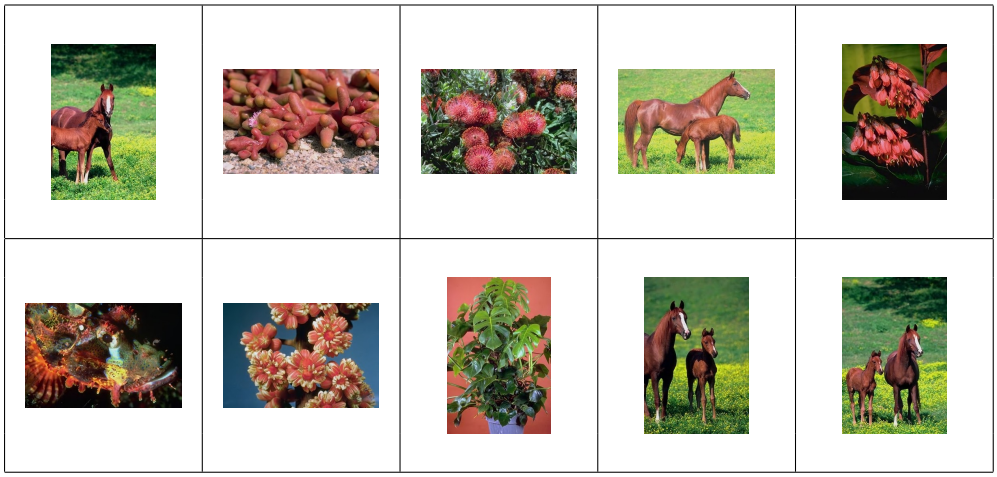


Figure 4.16: Example selected components query from COREL with top 10 results. Horses components retrieve brown things.

4.4 Textual and multimodal search

This section demonstrates the usefulness of exploiting textual information for multimedia retrieval on the TRECVID2002 and TRECVID2003 collections. The COREL3892 and COREL390 datasets are not used to evaluate textual search, since there is not much text available with these and an important part of the text is the class label. Doing text retrieval on this collection would be somewhat artificial.

Query:

Results:

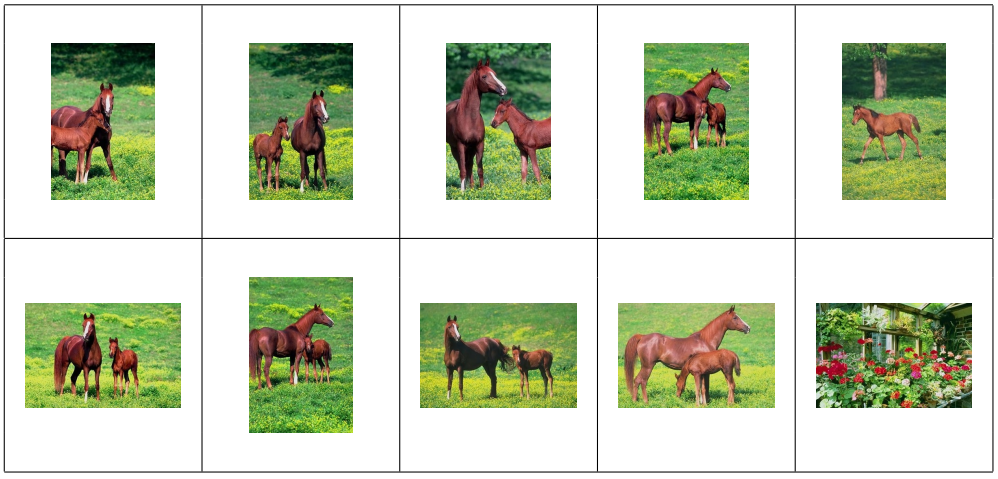


Figure 4.17: Example selected components query from COREL with top 10 results. Grass components retrieve horses.

4.4.1 Text only results

For modelling textual information we used the hierarchical language models as described in Section 3.6.4; scenes are assumed to be sequences of five consecutive shots. In the remainder of this section, the details for the two text collections are filled in and results are discussed.

Trecvid2002

The TRECVID2002 collection is stemmed and stopped and hierarchical language models are estimated for each document. For constructing queries from the topic descriptions, two different approaches are taken. The first set of textual queries, *Tshort*, is constructed by taking the text description from the topic. In the second set of queries, *Tlong*, the short queries are extended with the text from the speech transcripts of the video examples in the topic (if any). The assumption is that relevant shots share a vocabulary with example shots, thus using example transcripts may improve results. For both sets of queries we apply the same pre-processing as used for the collection, i.e., stopwords are removed and terms are stemmed. The results for the two variants are listed in Table 4.12 and show that the long queries outperform the short ones. It seems that indeed the transcripts of video examples and relevant shots share a vocabulary. However, the improvement in scores can be partly attributed to the fact that some of the video examples are taken directly from the search collection. Obviously, with long queries these shots are found and higher scores can be expected. When the four topics for which examples are taken from the test collection are removed from the topic set, the scores for the long queries drop from .1212 to .1082.

Trecvid2003

For the TRECVID2003 collection queries are constructed by manually selecting the content words from the topic descriptions. For example, the description of VT0122 (*Find shots of one or more cats. At least part of both ears, both eyes and the mouth must be visible. The body can be in any position*), is reduced to the query *cats*. Again stemming and stopping are applied to both queries and collection. No experiments with long queries are carried out on this collection. One reason for this is to avoid the influence of the exact matches that contaminate the TRECVID2002 results. Furthermore, short keyword type queries are more realistic. The average number of words people use in search engines is between 2 and 3. The results per topic for the short keyword queries on the TRECVID2003 collection are listed in Table 4.12

Table 4.12: Average precision per topic for text runs on TRECVID2002 and TRECVID2003 test collections.

TRECVID2002		
Topic	Tshort	Tlong
vt0075	.0000	.0082
vt0076	.4075	.6242
vt0077	.1225	.5556
vt0078	.1083	.2778
vt0079	.0003	.0006
vt0080	.0000	.0000
vt0081	.0154	.0333
vt0082	.0080	.0262
vt0083	.1669	.1669
vt0084	.7500	.7500
vt0085	.0000	.0000
vt0086	.0554	.0676
vt0087	.0591	.0295
vt0088	.0148	.0005
vt0089	.0764	.0764
vt0090	.0229	.0473
vt0091	.0000	.0000
vt0092	.0627	.0687
vt0093	.1977	.1147
vt0094	.0232	.0252
vt0095	.0034	.0021
vt0096	.0000	.0000
vt0097	.1002	.0853
vt0098	.0225	.0086
vt0099	.0726	.0606
MAP	.0916	.1212

TRECVID2003	
Topic	keywords
vt0100	.0070
vt0101	.0053
vt0102	.0527
vt0103	.2728
vt0104	.0074
vt0105	.3550
vt0106	.2845
vt0107	.1208
vt0108	.1191
vt0109	.0974
vt0110	.0110
vt0111	.0025
vt0112	.1769
vt0113	.0230
vt0114	.2974
vt0115	.0266
vt0116	.5564
vt0117	.0465
vt0118	.0002
vt0119	.1481
vt0120	.1760
vt0121	.0404
vt0122	.0181
vt0123	.3057
vt0124	.0885
MAP	.1296

One thing that can be learnt from the textual results on both collections is that visual information retrieval still has a long way to go. Even though the topics are designed for visual retrieval and the results are evaluated on their visual relevance, text retrieval outperforms visual retrieval by far.

4.4.2 Combining textual and visual runs

The previous sections treated visual and textual information separately, but it makes sense to combine them. One could imagine textual information setting the context (this shot is about Yasser Arafat), whereas visual information could filter the shots in the video where the person (with scarf) is actually visible. And vice versa, visual information could set a context (there is an object against a clear blue sky here), and textual information could help in deciding whether it is a helicopter, an aircraft or a balloon. Since both modalities are modelled in a probabilistic framework, combining them seems a viable option. In the framework, we can simply compute the joint probability of observing textual and visual part of a query.

$$\mathrm{p}(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathrm{p}(\mathcal{V}|\boldsymbol{\theta})\mathrm{p}(\mathcal{T}|\boldsymbol{\phi}) \tag{4.1}$$

Note that this requires two independence assumptions:

1. Textual terms and visual samples are generated independently: $\mathrm{p}(\mathcal{V}, \mathcal{T}|\cdot) = \mathrm{p}(\mathcal{V}|\cdot)\mathrm{p}(\mathcal{T}|\cdot)$.
2. The generation of documents in one modality is independent of the other modality. The generation of textual terms only depends on the language model and the generation of visual terms only on the visual model: $\mathrm{p}(\mathcal{V}|\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathrm{p}(\mathcal{V}|\boldsymbol{\theta})$ and $\mathrm{p}(\mathcal{T}|\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathrm{p}(\mathcal{T}|\boldsymbol{\phi})$.

Treating textual and visual information independently, contradicts the assumption that textual information is useful for visual multimedia retrieval. If textual information can actually help in retrieving relevant visual images or shots, then documents that have a high likelihood based on textual information should be more likely to be visually relevant than documents with a low textual score. Clearly, textual and visual information are dependent. As soon as a document is likely to be relevant based on the textual information, then the likelihood of observing something visually similar to the query examples should increase. For example, if the name Yasser Arafat is mentioned, the likelihood of observing him increases. Theoretically, this might lead to overly high scores for documents that match on both textual and visual information. Still, to keep the model simple, the independence assumptions listed above are used.

Table 4.13: Mono-modal and combined results (MAP) on TRECVID2003 for different types of queries

Query Type		Textual	Visual	Combined
Full	AllEx	.1296	.0281	.1428
Full	DesEx	.1296	.0245	.1341
Region	AllEx	.1296	.0264	.1416
Region	DesEx	.1296	.0217	.1342

Table 4.13 shows the results on the TRECVID2003 test collection for the combined visual and textual runs and for different types of queries (all, designated, full and regions). The mono-modal scores are repeated for clarity. The results show that indeed combining modalities is useful, the combined runs have consistently higher scores than the corresponding single modality runs. However, from looking at individual topics, we learnt that this effect is not due to a combination effect like sketched at the beginning of this section. Only for very few topics, both modalities contribute relevant documents to the combined run. Figure 4.18 shows an example *beadplot*. In this plot, each row represents the top N retrieved documents for one run. Relevant document within this top N are assigned a colour code. Documents that are retrieved by multiple runs are represented using the same colour code in each run.[11] The plot in Figure 4.18 shows that both the textual run (ASR) and the visual run (desEx) contribute results to the combination (desEx+ASR). For most queries however, it is the case that one modality is significantly better and dominates the combined run. An example of this is shown in Figure 4.19. Here, only the textual run (ASR) contributes relevant documents. Combining textual and visual runs in such cases will not yield better results than using the dominating modality. Still the inferior modality hardly distorts results. A combined runs avoids having to predict the best modality for a given topic and therefore is useful.

4.5 Discussion

This chapter showed how the generative probabilistic models can be used for ad hoc retrieval from generic multimedia collections.

One of the most important findings is that smoothing is crucial for the success of these models in a retrieval task. Smoothing deals with zero-frequency

[11] Beadplots are created using NIST's Beadplot tool, see `http://www.itl.nist.gov/iaui/894.02/projects/beadplot/`

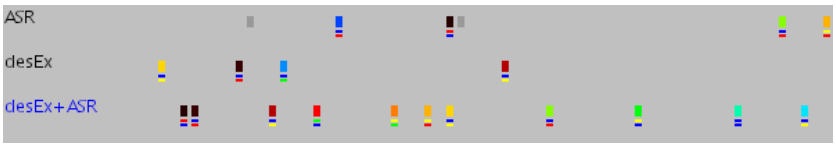


Figure 4.18: Visualisation of top N retrieved documents for a ASR run, designated example visual run and the combination of the two runs, for the road with vehicles query (VT0115).

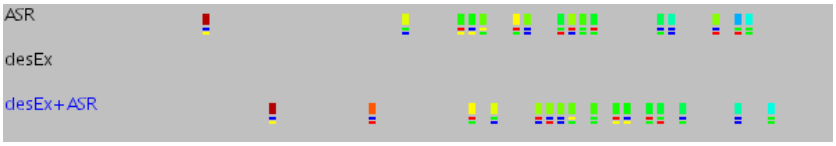


Figure 4.19: Visualisations (beadplot) of the top N retrieved documents for a ASR run, designated example visual run and the combination of the two runs, for the Mercedes logo query (VT0108).

problems and explains away common query terms. For text retrieval, the importance of the technique has been known. For image retrieval, the result is new.

Experiments with the TRECVID collections showed that it is hard to predict in advance which visual examples will give good results. Simply using all available visual examples as if they are one large bag of samples results in higher scores than a priori selecting a single representative example. However, running separate queries for each example and combining the results afterwards in a round-robin fashion, is better still. This supports the intuition that example sets represent OR type queries; the request is for something that looks like example A OR example B OR example C, but not necessarily like all of them.

The Gaussian mixture models approach naturally allows for local queries. All feature vectors are computed locally on a small block of pixels. A simple selection of a subset of these blocks in a query image suffices to zoom in on specific image parts. The soft clustering that the components induce on an image allows for selecting irregularly shaped regions that are coherent in colour, texture and/or location. When a single coherent region is used as a query, results are often intuitive. Although this does not necessarily translate to better retrieval results, intuitive results are useful. It is important that a user understands why document are retrieved and how the query can be adjusted for better results.

The experiments with ASR transcripts show that text is still a valuable source of information for disclosing multimedia material. On average, textual

results are significantly better than visual results. Still, for some queries a visual run outperforms a textual run. Moreover, in a multimodal run, the better of the two modalities per topic dominates the results, resulting in an average score that is higher than either of the mono-modal scores. For a particular topic, choosing a specific modality may give higher scores, but on average, a combination works best.

5

Model extensions and alternative uses

In this chapter alternative ways of using the generative probabilistic models from Chapter 3 are introduced. Each of the Sections 5.2–5.6 treats a different variant and can be read in isolation. Section 5.1 introduces the glue in this chapter: a common probabilistic framework that accommodates all variants. Section 5.2 reverses the generation process from Chapter 3 by generating document samples from query models. Section 5.3 introduces a new way of estimating document models from data that captures how a given document differs from the average document. Section 5.4 develops a Bayesian extension of the models that incorporates the uncertainties in model estimation. Section 5.5 shows how the generative models can be used in a cross-modal setting. Section 5.6 discusses strategies for making query processing more efficient. Finally section 5.7 summarises the chapter by showing how the different variants fit in the common framework.

Both the document sample generation approach of Section 5.2 and the model estimation techniques of Section 5.3 have been published before (Westerveld and De Vries, 2004). Early ideas for relating textual terms and visual documents (cf. Section 5.5) are presented in (Westerveld et al., 2000). Finally, experiments with the Asymptotic Likelihood Approximation (Section 5.6.2) have been published as part of (Westerveld et al., 2003b).

5.1 Generative models and relevance

Section 3.7 explained that the models used so far take a classification view on retrieval. The models of the documents in a collection are seen as generating sources, and the query is assumed to be an observation from one of these.

As pointed out in that section, this is a somewhat controversial view. The main argument against it is that the notion of relevance is ignored in the models. This section places the generative models in a general probabilistic retrieval framework that includes relevance. This way, the assumptions made in the generative models with regard to relevance will be made explicit. Section 5.1.1 introduces the framework, Section 5.1.2 shows how the generative models from Chapter 3 fit in and Section 5.1.3 presents an alternative reading of the framework.

5.1.1 Probabilistic framework

Although Maron and Kuhns (1960) were the first to consider probability theory for information retrieval, the binary independence retrieval model (Robertson and Sparck Jones, 1976) was the first probabilistic approach that was actually put to use. To date, their 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 document representation (see also Section 2.1.2). In a more recent paper Sparck Jones et al. (2000) present this classical probabilistic model starting from the "basic question":

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

Lafferty and Zhai (2003) start from the same basic question to show this classical model is probabilistically equivalent to the modern language models for information retrieval. This section follows Lafferty and Zhai to show how the generative models from Chapter 3 relate to the classical probabilistic models.

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 $\mathrm{P}(r|D,Q)$.[1] This can be estimated indirectly using Bayes' rule:

$$\mathrm{P}(r|D,Q) = \frac{\mathrm{P}(D,Q|r)\mathrm{P}(r)}{\mathrm{P}(D,Q)} \tag{5.1}$$

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

$$\frac{\mathrm{P}(r|D,Q)}{\mathrm{P}(\bar{r}|D,Q)} = \frac{\mathrm{P}(D,Q|r)\mathrm{P}(r)}{\mathrm{P}(D,Q|\bar{r})\mathrm{P}(\bar{r})}. \tag{5.2}$$

[1] As in the previous chapters, random variables are omitted when instantiated, unless this may cause confusion. Thus $\mathrm{P}(r|D,Q)$ means $\mathrm{P}(R = r|D,Q)$.

As Lafferty and Zhai (2003) 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.

5.1.2 Query generation framework

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:

$$\frac{\mathrm{P}(r|D,Q)}{\mathrm{P}(\bar{r}|D,Q)} = \frac{\mathrm{P}(D,Q|r)\mathrm{P}(r)}{\mathrm{P}(D,Q|\bar{r})\mathrm{P}(\bar{r})} = \frac{\mathrm{P}(Q|D,r)\mathrm{P}(D|r)\mathrm{P}(r)}{\mathrm{P}(Q|D,\bar{r})\mathrm{P}(D|\bar{r})\mathrm{P}(\bar{r})} = \frac{\mathrm{P}(Q|D,r)\mathrm{P}(r|D)}{\mathrm{P}(Q|D,\bar{r})\mathrm{P}(\bar{r}|D)} \tag{5.3}$$

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

Assumption 5.1.1 *$P(Q, D|\bar{r}) = P(Q|\bar{r})P(D|\bar{r})$,*

$\mathrm{P}(Q|D, \bar{r})$ 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:

$$\textsc{rsv}_{\text{Qgen}}(d) = \mathrm{P}(q|d,r)\frac{\mathrm{P}(r|d)}{\mathrm{P}(\bar{r}|d)} \tag{5.4}$$

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 might be more likely to be relevant than graphic images, CNN videos might 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 (Kraaij et al., 2002). 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

$$\textsc{rsv}_{\text{Qgen}}(d) = \mathrm{P}(q|d,r) \tag{5.5}$$

The language models and the Gaussian mixture models from Chapter 3 are special cases of this query generation variant. For text, $\mathrm{P}(Q|D, r)$ is estimated as the probability of observing Q from the (smoothed) language model of D. For images, this probability is estimated as the probability of observing the visual samples in the query from the Gaussian mixture model of the document:

$$\textsc{rsv}_{\text{Qgen}}(d) = \mathrm{P}(q|d,r) \equiv \prod_{\boldsymbol{v}\in\mathcal{V}_q} \kappa \mathrm{p}(\boldsymbol{v}|\boldsymbol{\theta}_d) + (1-\kappa)\mathrm{p}(\boldsymbol{v}) \tag{5.6}$$

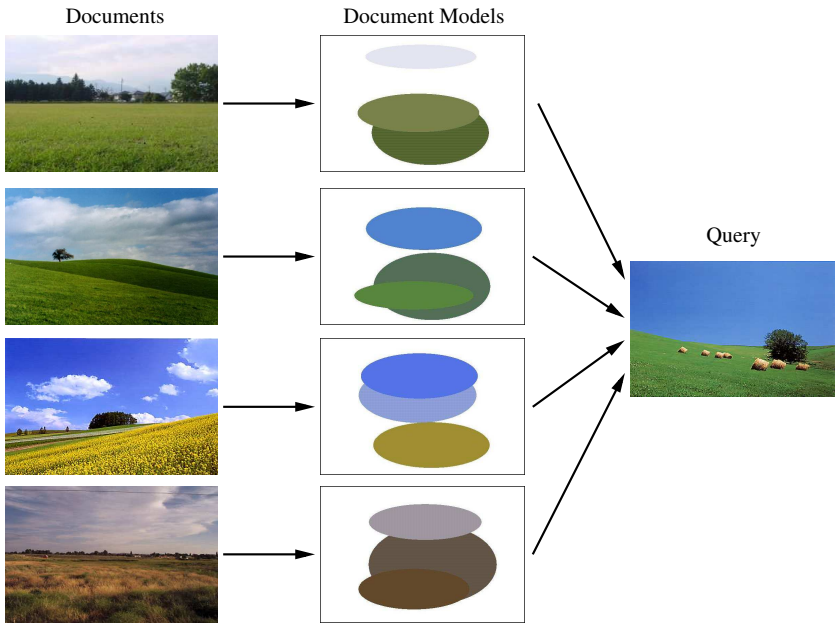


Figure 5.1: Visualisation of query generation framework.

Summarising, the query generation approach builds a generative model for each document in the collection. The likelihood of observing the query from each as these models is used for ranking. Figure 5.1 visualises this for the visual models.

5.1.3 Document generation framework

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

$$\frac{\mathrm{P}(r|D, Q)}{\mathrm{P}(\bar{r}|D, Q)} = \frac{\mathrm{P}(D, Q|r)\mathrm{P}(r)}{\mathrm{P}(D, Q|\bar{r})\mathrm{P}(\bar{r})} = \frac{\mathrm{P}(D|Q, r)\mathrm{P}(Q|r)\mathrm{P}(r)}{\mathrm{P}(D|Q, \bar{r})\mathrm{P}(Q|\bar{r})\mathrm{P}(\bar{r})} \tag{5.7}$$

Under Assumption 5.1.1, and ignoring all factors independent of D, we arrive at the following RSV:

$$\mathrm{RSV}_{\mathrm{Dgen}}(d) = \frac{\mathrm{P}(d|q, r)}{\mathrm{P}(d|\bar{r})} \tag{5.8}$$

This document generation variant is the one used in the binary independence retrieval model (Robertson and Sparck Jones, 1976; Sparck Jones et al., 2000), although the dependence on Q is implicit there. They estimate probabilities based on term distributions in relevant and irrelevant documents. For Gaussian mixture models, estimation of $\mathrm{P}(D|Q, r)$ and $\mathrm{P}(D|\bar{r})$, is treated in the next section.

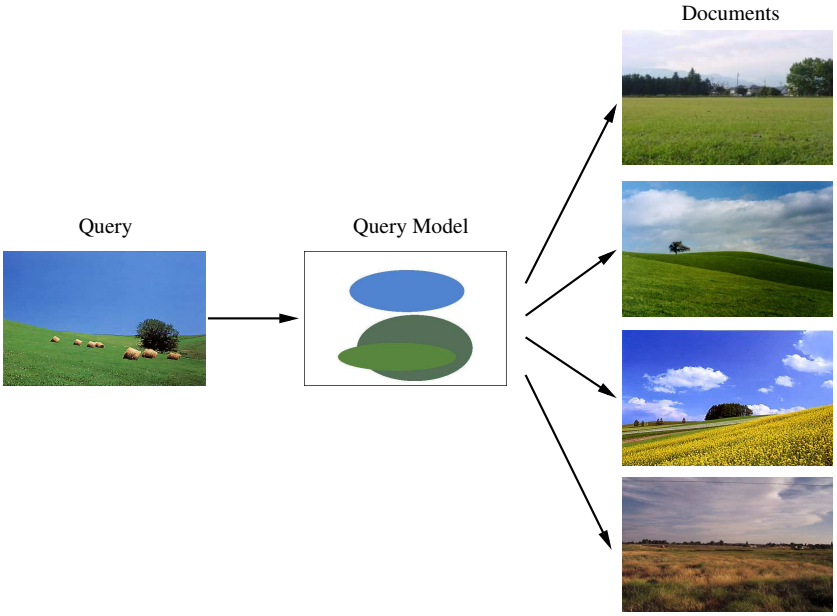


Figure 5.2: Visualisation of document generation framework.

5.2 Document generation

The previous section introduced a general probabilistic framework. The approach from the previous chapters, building a model for each document and computing the query likelihood, is an instantiation of the query generation variant (Section 5.1.2). This section implements and evaluates the document generation variant for Gaussian mixture models. The document generation variant essentially reverses the process: a model is built from the query and the likelihood of the document samples is computed for each of the documents in the collection (see Figure 5.2).

One reason to exploit this direction is the fact that it could possibly solve the problems with the OR functionality of multiple example queries (see Section 4.3.2). The query generation variant tries to retrieve document models that explain *all* query examples. We have used tricks to combine multiple examples in an OR-like fashion (round-robin combination), but basically the retrieval framework so far facilitates AND-querying only. If however the process is reversed and topic models are built to generate document samples, then the different components in a topic model could capture different aspects of the (multiple example) topic and document samples could possibly be explained by only a subset of the topic model components. Section 5.2.1 shows how the Gaussian mixture models can be used in the document generation approach. Section 5.2.2 evaluates the approach and compares it to

the query generation variant.

5.2.1 Document generation with Gaussian mixture models

To implement a document generation variant, two probability distributions have to be estimated (cf. Equation 5.8): $\mathrm{P}(D|Q,r)$ and $\mathrm{P}(D|\bar{r})$. In analogy to the query generation approach, we estimate $\mathrm{P}(D|Q,r)$ as the probability that the set of document samples $\mathcal{V}_d$ is generated from the query model $\boldsymbol{\theta}_q$. The probability of a document conditioned on the unrelevant event is estimated as the joint background density of the document samples. Thus, to rank documents in the document generation variant, we use

$$\textsc{rsv}_{\mathrm{Dgen}}(d) = \frac{\mathrm{P}(d|q,r)}{\mathrm{P}(d|\bar{r})} \equiv \frac{\mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta}_q)}{\mathrm{p}(\mathcal{V}_d)} \tag{5.9}$$

In the query generation approach, the maximum likelihood estimates $\mathrm{p}(\cdot|\boldsymbol{\theta})$ have been smoothed to explain away common query samples. The specific smoothing technique used was interpolation with a more general background model (cf. Section 3.6). The technique proved crucial for successful retrieval (see Chapter 4). Therefore, in the document generation approach the same technique is applied and we redefine $\textsc{rsv}_{\mathrm{Dgen}}$ as:

$$\textsc{rsv}_{\mathrm{Dgen}}(d) \equiv \frac{\prod_{\boldsymbol{v}\in\mathcal{V}_d} \kappa\mathrm{p}(\boldsymbol{v}|\boldsymbol{\theta}_q) + (1-\kappa)\mathrm{p}(\boldsymbol{v})}{\prod_{\boldsymbol{v}\in\mathcal{V}_d} \mathrm{p}(\boldsymbol{v})} \tag{5.10}$$

$$= \prod_{\boldsymbol{v}\in\mathcal{V}_d} \left[\frac{\kappa\mathrm{p}(\boldsymbol{v}|\boldsymbol{\theta}_q)}{\mathrm{p}(\boldsymbol{v})} + (1-\kappa)\right]. \tag{5.11}$$

In Equation 5.11, the *idf* function of smoothing is apparent; common document samples contribute less to the RSV (cf. Section 3.6 and Equation 3.22).

In the query generation approach, the background model was estimated by marginalisation over either all models in the document collection, or all models in a reference collection. In document generation, we ignore document models and usually only treat a single query model at a time. Therefore, a separate reference collection is needed to estimate the background density $\mathrm{p}(\boldsymbol{v})$.

5.2.2 Document generation experiments

The TRECVID2003 collection (see Section 4.1.1) is used to compare the new document generation variant to the query generation variant discussed in

Chapter 3. For the document generation variant, topic models are built from two different sets of query samples. The first set contains all available query samples, the second only those from manually selected *interesting* regions (see Section 4.3.3). Since the focus in this section is on multiple example queries, both sets consist of samples from all available query examples. The RSV for each document is computed as the likelihood of the set of document samples (Equation 5.11). The background probabilities are estimated over a small (1%) random sample from the comparable development set, available with the TRECVID2003 collection.

In the query generation variant, a document model is built for each document in the collection. Documents are ranked using their likelihood of generating sets of query samples (Equation 5.6). The sets of query samples are the same ones used in the document generation variant, i.e., full images, or samples from manually selected regions.

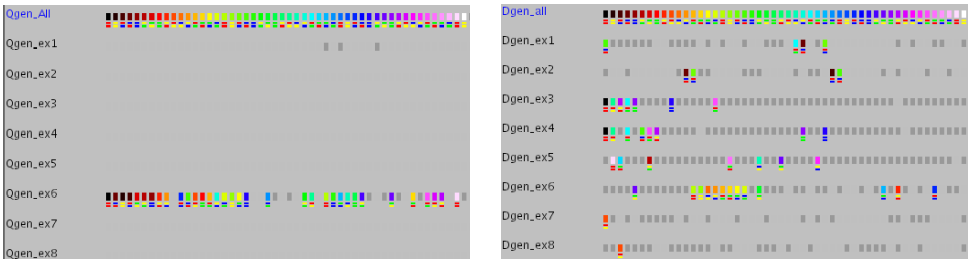
In total, four variants are tested; two model variants (document and query generation), each in combination with two sets of query samples (full and regions). Each of these variants is evaluated in isolation, as well as in combination with textual information. The textual information follows a query generation approach like in Section 4.4.1.[2] To combine visual and textual information, we compute the joint probability of textual terms and visual samples, like in Section 4.4.2.

Table 5.1 shows the results for the different settings. For full image examples, query generation outperforms document generation. For selected regions, this is reversed. The reason for this becomes apparent when individual topics are studied rather than only mean scores. 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 results. Figure 5.3 (left) shows this effect. Each row of these beadplots shows the top N retrieved documents for a given run. Documents that are within the top 50 results 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. In the document generation approach, the topic models represent important common aspects of the query examples. Thus, all examples contribute to the combined result (see Figure 5.3,

[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.

Table 5.1: MAP scores for different system variants. Both the scores for using visual information only (MAP_{vis}) and the scores for a combination of visual and textual information (MAP_{MM}) are listed (The MAP for textual only is .130).

Model	Qsamples	MAP_{vis}	MAP_{MM}
Qgen	full	.028	.143
Qgen	region	.026	.142
Dgen	full	.026	.119
Dgen	region	.026	.167



Query Generation Document Generation

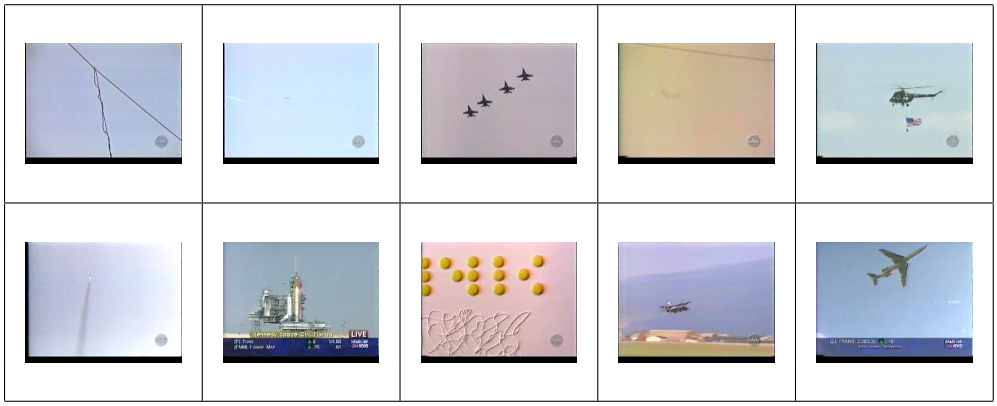
Figure 5.3: Visualisations (beadplots) of the top N retrieved documents for the multiple-example run (top row) and the individual example runs, for a rocket launch query (VT0107).

right) and more generic matches are found. Clearly, the AND combination of examples in the query generation approach does not work. As stated before, it is unlikely that a single model can explain all examples well. Figure 5.3 shows that indeed the best models in this approach do not explain all examples, but they all explain the same example. The highest ranked documents in the document generation approach each explain different examples, thus a document generation strategy, is indeed closer to an OR-strategy. It retrieves documents that are explained by any of the example models.

The fact that the query generation approach is mainly good at identifying near exact matches explains why selecting regions harms results there. A near exact match is similar both in foreground and in background. When regions are selected, there is simply less data to match on. In the document generation approach however, for the full image example set, the topic models include a mixture of different (unrelevant) backgrounds. When regions are selected, the models become more focused on the actual query and ignore the background. This means the document generation approach could potentially benefit from selecting regions and thus find more generic matches. This effect only becomes apparent in combination with textual information.

In the visual only setting, the full and region based set have the same scores for the document generation variant. In combination with textual information, however, the region based document generation approach is better than any query generation variant. Apparently, the visual information can provide a generic visual context, while the textual information zooms in on specific results. For example, for topics that ask for aeroplanes, helicopters or rocket launches, the visual model captures the fact that we are looking for objects against a background of sky. The textual information can then help distinguishing between the specific objects (Figure 5.4).

Visual only results:



Visual+text results:

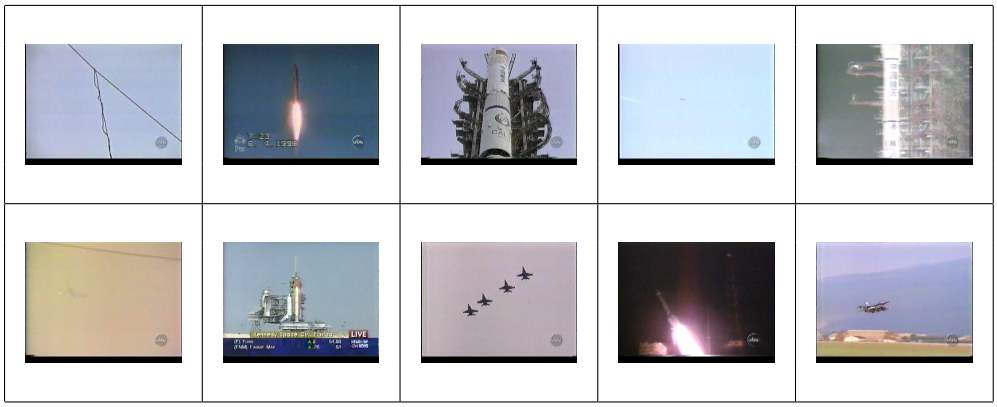


Figure 5.4: Document generation results for Rocket launch query (topic107). The visual information sets the context (top rows, sky background) adding textual information fills in specifics (bottom, rockets)

5.2.3 Document generation versus query 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.

$$\text{RSV}_{\text{Dgen}}(d) = \prod_{\boldsymbol{v}\in\mathcal{V}_d}\left[\frac{\kappa \text{p}(\boldsymbol{v}|\boldsymbol{\theta}_q)}{\text{p}(\boldsymbol{v})} + (1-\kappa)\right] \leq \prod_{\boldsymbol{v}\in\mathcal{V}_d}\max_{v'}\left[\frac{\kappa \text{p}(\boldsymbol{v'}|\boldsymbol{\theta}_q)}{\text{p}(\boldsymbol{v'})} + (1-\kappa)\right] \tag{5.12}$$

Thus, the (hypothetical) document that is a repetition of the single most likely value 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 the background probability. 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 results show that indeed, in a mono-modal setting, query generation gives better results than document generation (Table 5.1). However, combined with textual information, document generation outperforms query generation when the query models are built from manually 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. 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 and relates to the effect of setting the visual context discussed before. Highly ranked documents may show only the most likely of the query aspects (e.g. *sky*), but the textual information can then help to re-rank the results, or to zoom in on relevant documents (e.g., *rockets*).

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 the best approach. However, KL is not analytically solvable for Gaussian mixture models. Section 5.6.2 discusses an approximation that has been proposed.

5.3 Smoothing during training

The experimental results presented in Chapter 4 have confirmed that smoothing is an important prerequisite for the success of generative probabilistic models in retrieval. The intuition behind this is that query samples which are common are down-weighted, decreasing their influence on retrieval results. The same idea can be applied when estimating the models' parameters. It is possible to decrease the influence of common samples on the parameter estimation for an image model. Gaussian mixture models built this way will mainly describe how a given image differs from an average image. The hypothesis is that this will lead to better image models, since no parameters are wasted on describing commonalities. Recently, the same ideas are proposed for text retrieval in (Sparck Jones et al., 2003), where so-called parsimonious language models are developed. Parsimonious models are mixtures of language models, where each model describes a different level of specificity. The models have been applied in (Hiemstra et al., 2004). The following subsection describes how the same ideas can be applied to image model estimation.

5.3.1 EM for interpolated estimates

To pursue the idea of building more focused image models that describe typicalities rather than commonalities, background probabilities have to be incorporated in the training process. Like in the standard EM algorithm (Section 3.5.1), hidden variables h_{ij} indicate the assignment of samples $\boldsymbol{v_j}$ to components c_i. In the new training variant however, samples can not only be assigned to one of the model components, but also to the background. The assignment of sample $\boldsymbol{v_j}$ to the background model is indicated by $h_{\mathrm{BG}j}$. One way of looking at this, is as if the background is just another component and a larger mixture model has to be trained. The new mixture model, is a mixture of C components like we had before and a special component for

the background model. The background component differs from the others in two ways. First, it has a different distribution. While the other components have Gaussian distributions, the background model is computed by marginalising over a set of models (see Section 3.6.3); it has a Gaussian mixture distribution. Second, it is the same for all documents in the collection. While each of the normal components is document specific, the background model describes the general distribution in images and therefore is the same for all documents.

The EM-algorithm can be applied as before. The E-step changes to:

$$h_{ij} = \mathrm{P}(c_i|\boldsymbol{v_j}) = \frac{\mathrm{p}(\boldsymbol{v_j}|c_i)\mathrm{P}(c_i)}{\sum_{c=1}^{C} \mathrm{p}(\boldsymbol{v_j}|c_c)\mathrm{P}(c_c) + \mathrm{p}(\boldsymbol{v_j})\mathrm{P}(\mathrm{BG})} \tag{5.13}$$

$$h_{\mathrm{BG}j} = \mathrm{P}(\mathrm{BG}|\boldsymbol{v_j}) = \frac{\mathrm{p}(\boldsymbol{v_j})\mathrm{P}(\mathrm{BG})}{\sum_{c=1}^{C} \mathrm{p}(\boldsymbol{v_j}|c_c)\mathrm{P}(c_c) + \mathrm{p}(\boldsymbol{v_j})\mathrm{P}(\mathrm{BG})}, \tag{5.14}$$

where $\mathrm{P}(\mathrm{BG}|\boldsymbol{v_j})$ is the posterior probability that $\boldsymbol{v_j}$ is from the background and $\mathrm{P}(\mathrm{BG})$ is the prior probability that background samples are observed under the current model.

In the M-step the component parameters $\boldsymbol{\mu}_i$,$\boldsymbol{\Sigma}_i$ and $\mathrm{P}(c_i)$ are updated like in Equations 3.13–3.15 (repeated here for completeness). In addition, the background prior for the current model $\mathrm{P}(\mathrm{BG})$ needs updating.

$$\boldsymbol{\mu_i}^{\mathrm{new}} = \frac{\sum_j h_{ij}\boldsymbol{v_j}}{\sum_j h_{ij}}, \tag{5.15}$$

$$\boldsymbol{\Sigma_i}^{\mathrm{new}} = \frac{\sum_j h_{ij}(\boldsymbol{v_j} - \boldsymbol{\mu_i}^{\mathrm{new}})(\boldsymbol{v_j} - \boldsymbol{\mu_i}^{\mathrm{new}})^T}{\sum_j h_{ij}}, \tag{5.16}$$

$$\mathrm{P}(c_i)^{\mathrm{new}} = \frac{1}{N}\sum_j h_{ij} \tag{5.17}$$

$$\mathrm{P}(\mathrm{BG})^{\mathrm{new}} = \frac{1}{N}\sum_j h_{\mathrm{BG}j} \tag{5.18}$$

The background model $\mathrm{p}(\boldsymbol{v})$ has been defined as a marginalisation over a reference collection (cf. Equation 3.24). This reference collection can be either the collection for which we are estimating models, or a comparable collection. The idea is that the background model has high probability for common terms. However, in the new EM variant, we are estimating a collection of models that does not describe commonalities. Therefore, marginalising over this collection will not give us accurate estimates for background probabilities and a comparable collection is needed.

Figure 5.5: Sphinx example. Original image (left) and samples selected by EM algorithm (right).

Using the EM variant outlined above, common samples will be assigned to the background and only distinguishing samples will be used in estimating the components' parameters. Figure 5.5 shows an example image (left) along with its assignment image (right), depicting the samples that are, after convergence, assigned to one of the model components, i.e., not assigned to the background. In the assignment image, the transparency of a sample $\boldsymbol{v_j}$ is proportional to $\mathrm{P}(\mathrm{BG}|\boldsymbol{v_j})$. So, fully transparent samples are completely assigned to the background. What remains visible is the proportion of the samples that the model parameters are estimated from.

5.3.2 Background trained models and retrieval

Once the models have been trained using these background probabilities, there are two ways to use them. First, the model specific background priors $\mathrm{P}(\mathrm{BG})$ can be used as the mixing parameters for smoothing, replacing $1-\kappa$ in Equation 3.23:

$$\mathrm{p}(\boldsymbol{v}|\boldsymbol{\theta}) = (1-\mathrm{P}(\mathrm{BG}))\left[\sum_{i=1}^{C}\mathrm{P}(c_i|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{v}|c_i,\boldsymbol{\theta})\right] + \mathrm{P}(\mathrm{BG})\mathrm{p}(\boldsymbol{v}). \qquad (5.19)$$

This *document dependent smoothing* approach explicitly captures and uses the fact that the model has learnt the proportion of an image explained by the background model. It also means we are implicitly looking for document images that have the same proportion of common samples as the query image.

Alternatively, we can ignore the trained background priors and simply use the trained models as before with a fixed mixing parameter κ. To do so, we need to re-normalise the components priors $\mathrm{P}(c_i)$ to sum to 1. Still, returning to a fixed κ does not mean the models are the same as in Chapter 3. This *document independent smoothing* approach treats the newly trained models as better, more focused models, that explain how a document differs from the collection. In retrieval (assuming a query generation approach), common query samples can be explained from the background model and

training method	Qgen	Dgen
original	.028	.026
BGtrain with model independent smoothing	.018	.034
BGtrain with model dependent smoothing	.011	.034

Table 5.2: Mean average precision scores for background trained models and original models, for both query generation and document generation approaches.

thus are down-weighted. This reduced contribution is however document independent. It does not depend on the amount of common samples that can be expected from the document model.

5.3.3 Experiments

To compare the background trained models to the original models, we use the TRECVID2003 collection (see Section 4.1.1). Both document dependent and document independent smoothing are tested and everything is evaluated on both query and document generation approaches. In each setting, all available visual examples from a query are used. The background probabilities $p(\boldsymbol{v})$ that are used during training are estimated on a small (1%) sample from a comparable collection (the TRECVID2003 development set).

Table 5.2 shows the results. In the document generation approach, it helps to focus on modelling distinguishing aspects of images, the models trained using background probabilities outperform the original models. In fact the obtained scores .034 are higher than for any other document or query generation approach that uses only visual information.

In the query generation approach, however, using background probabilities during training harms the results. A possible explanation is that near exact matches, the kind of matches the query generation approach tends to find, are less likely. When background probabilities are used during training, the document models no longer describe the full document, thus there is simply less data to match on. Another explanation can perhaps be found in the value of the mixing parameter κ. When document models do not contain background information, all common samples in the query need to be explained from the background model. Thus perhaps the influence of the background model needs to be increased when background probabilities are used during training, i.e., perhaps a smaller value for κ should be used in these cases.

The difference between document dependent and document independent

smoothing can also be explained from this *idf* role of smoothing. Normally, smoothing reduces the effect of common query samples on the results. The reason for this is that *all* documents in the collection will have a high background score for these samples, thus the contribution of the individual foreground scores is relatively small. Now that we have document dependent smoothing parameters, some documents will get a high score whilst others get low scores for these common samples (depending on the document models' background priors). Thus, in the document dependent setting, common samples partly decide which documents are interesting and the *idf* effect of smoothing is reduced. The document generation approach does not suffer from this, since the mixing weights remain document independent. Although κ is dependent on the background prior P(BG) of the query model, it is the same for all documents in the collection. Therefore, the background probability of a sample remains document independent. As we have already seen in Section 4.2.3, Figure 4.7, the exact choice of κ is not very important, as long as it is somewhere between, 0.1 and 0.9. For most queries, if not all, this will be the case, thus using query specific κs hardly influences the results.

5.4 Bayesian extensions

So far, the probability of a visual query example conditioned on a document and the relevant event has been estimated as $\mathrm{P}(Q|D,r) \equiv \mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta}_d)$, i.e., the probability of observing the query samples from the document model.[3] Here, the document model is a single point estimate, namely the maximum likelihood estimator of the document samples $\boldsymbol{\theta}_d = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta})$.[4] This approach does not take into account the amount of uncertainty in $\boldsymbol{\theta}_d$. This section develops a Bayesian approach that does take this uncertainty into account by integrating out $\boldsymbol{\theta}$. In the process, the relation of the Gaussian mixture model to other generative models will become clear. Section 5.4.1 starts from the graphical representation of the models (cf. Section 3.3.2) and step by step drops assumptions to get closer to a Bayesian model. Section 5.4.2 explains how already estimated models can be used in a setting that is close to a full Bayesian approach, but does not require re-estimation of all parameters.

[3] In the document generation approach, this is reversed.

[4] For simplicity, smoothing is ignored in this section.

5.4.1 From maximum likelihood estimates to Bayesian approaches

In a Bayesian approach, instead of using single point estimates for the generating model, all possible models are considered. In Chapter 3 only a single model at a time was considered. This section starts from the maximum likelihood approach of Chapter 3 and moves step by step toward a Bayesian approach. Gaussian mixture models and language models are treated separately.

Gaussian mixture models

To develop a Bayesian variant of using the Gaussian mixture models, let us first make explicit the existence of multiple models in a collection. This will help us later to incorporate uncertainty about the generating model for a given document. The process for generating a collection of images, as visualised in Figure 5.6, is the same as the one introduced in Section 3.3.2 (Figure 3.5), but the process is repeated for each of the images in the collection.

For each of the M documents (images) in the collection,

1. pick the corresponding Gaussian mixture model $\boldsymbol{\theta}$.
2. For each sample $\boldsymbol{v}$ in the document,
   (a) pick a random component c_i from Gaussian mixture model θ according to the prior distribution over components $\mathrm{P}(c)$ and
   (b) draw a random sample from c_i according to the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$.

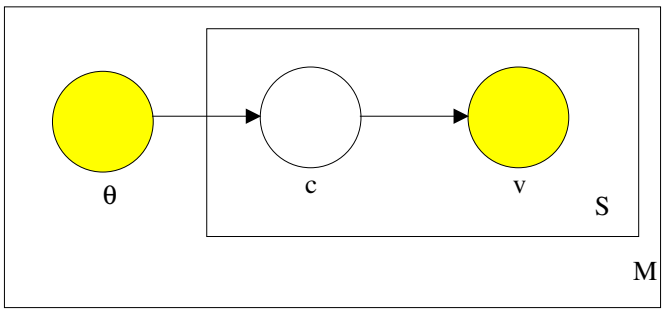


Figure 5.6: Graphical representation of Gaussian mixture model for collections.

Here, $\boldsymbol{\theta}$ is an observed variable. The mixture model from which the samples for a given document are drawn, is known. For a given sample however, it

is unknown which component generated it, thus components are unobserved variables. A collection of documents is described by a set of models $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$. Each of these has C components, thus in total, there are $C{\cdot}M$ hidden component variables to describe a collection. However, since each document is modelled by a separate Gaussian mixture, the components cannot be selected arbitrarily from the whole set of $C{\cdot}M$ components as suggested by Figure 5.6. The choice of components is restricted by the models $\boldsymbol{\Theta}$ in the following ways:

- For each model, at most C components c_i exist for which $\mathrm{P}(c_i|\boldsymbol{\theta}) > 0$
- Each component belongs to only one model:
  $\mathrm{P}(c|\boldsymbol{\theta}_j) > 0 \implies \forall_{k \neq j} \mathrm{P}(c|\boldsymbol{\theta}_k) = 0$

If these restrictions are dropped, thus allowing any component to be selected from any model, the model is equivalent to probabilistic latent semantic indexing (pLSI) (Hofmann, 1999). In pLSI, there exists a set of latent classes, or aspects, each describing a different topic. Each document defines a mixture over these classes and each class models a distribution over samples. Traditionally pLSI has been used for text retrieval, where $\mathrm{P}(\cdot|c)$ is a multinomial distribution over words. In this thesis, the observations are visual samples described by a feature vector and $\mathrm{p}(\cdot|c)$ is assumed Gaussian.

So far, we have used the Gaussian mixture models in retrieval under the assumption that query samples and document samples are drawn from the same, observed model. First, the document samples have been used to estimate the models parameters and then the probability that this model generates the query samples has been computed.

$$\boldsymbol{\theta}_d = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta}) \tag{5.20}$$

$$\mathrm{P}(q|d,r) \equiv \mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta}_d) \tag{5.21}$$

$$\tag{5.22}$$

Looking at the form of the main probability of interest, $\mathrm{P}(q|d,r)$, it is natural to replace the maximum likelihood estimate with the maximum a posteriori[5] estimate for $\boldsymbol{\theta}_d$:

$$\boldsymbol{\theta}_d = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d) = \arg\max_{\boldsymbol{\theta}} \frac{\mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{V}_d)} = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}) \tag{5.23}$$

[5]Often the term MAP is used as a shorthand for maximum a posteriori. In this thesis however, MAP stands for mean average precision and maximum a posteriori is written in its full form.

Here, $\mathrm{p}(\boldsymbol{\theta})$ is the prior distributions over the model parameters. It can model the degree of belief in different parameter settings. When all possible model parameters are considered equally likely a priori, the maximum likelihood and maximum a posteriori estimates are equivalent. The latter is introduced here, because of its similarity to the Bayesian approach. Like the Bayesian approach, it computes the likelihood of the query given the model ($\mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta})$) and the likelihood of the model given the document samples ($\mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d)$). Both maximum likelihood and maximum a posteriori approaches do not take into account the amount of uncertainty in the estimates for the model. In some cases, many models are very likely given the document samples, while in other cases there may indeed be a single outstanding model. These differences are not taken into account and only the single best estimate is considered.

Alternatively, a Bayesian approach considers all possibilities by integrating out the model parameters. This means the models $\boldsymbol{\theta}$ are viewed as unobserved variables and, in the generative process, it is as if we pick a random model from the prior distribution for each of the M documents in the collection, instead of the one corresponding to the document. Figure 5.7 shows the graphical representation of this Bayesian variant.

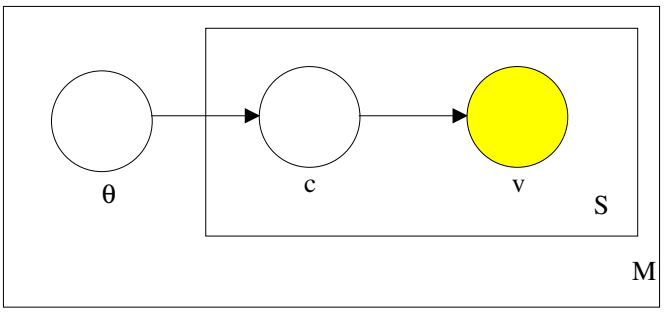


Figure 5.7: Graphical representation of LDA model.

To compute the likelihood for a set of samples $\mathcal{V}$, in the Bayesian approach the models are integrated out:

$$\mathrm{p}(\mathcal{V}) = \int_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{\theta}) \prod_{\boldsymbol{v}\in\mathcal{V}} \sum_{i=1}^{C} [\mathrm{P}(c_i|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{v}|c_i)]\,\mathrm{d}\boldsymbol{\theta}. \tag{5.24}$$

The resulting model is equivalent to the latent Dirichlet allocation (LDA) model developed by Blei et al. (2003). LDA is a latent variable model for generating documents. In the LDA model for each sample, a latent variable c_i is selected and a sample $\boldsymbol{v}$ is drawn from the model corresponding to that variable. The prior distribution over models $\boldsymbol{\theta}$ is assumed to be Dirichlet,

the natural conjugate for multinomial distributions.[6] LDA was originally developed for textual document collections, but has been extended for visual documents and multi-modal documents in (Blei and Jordan, 2003). Girolami and Kabán (2003) show the maximum a posteriori approximation of LDA is equivalent to pLSI.

To use the Bayesian approach (i.e., the LDA model) for retrieval, $\mathrm{P}(q|d,r)$ needs to be redefined in terms of the model. Like in the maximum a posteriori approach, we consider the query likelihood given the model ($\mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta})$) and the likelihood of the model given the document samples ($\mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d)$), but in the Bayesian variant the models are integrated out. This leads to

$$\mathrm{P}(q|d,r) \equiv \mathrm{p}(\mathcal{V}_q|\mathcal{V}_d) = \int_{\boldsymbol{\theta}} \mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d)\mathrm{d}\boldsymbol{\theta} \tag{5.25}$$

$$= \int_{\boldsymbol{\theta}} \frac{\mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta})\mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{V}_d)}\mathrm{d}\boldsymbol{\theta}. \tag{5.26}$$

Estimation of the model parameters ($\mathrm{P}(\cdot|c)$, $\mathrm{P}(c|\boldsymbol{\theta})$, $\mathrm{P}(\boldsymbol{\theta})$), is intractable, but can be solved using variational inference procedures. The interested reader is referred to (Blei et al., 2003) and (Blei and Jordan, 2003) for details. In this thesis, the focus is on using generative models that are estimated directly from documents, rather than from a whole collection. Section 5.4.2 shows how such models can be used in a Bayesian fashion.

Language models

In parallel to the Gaussian mixture discussion above, we derive Bayesian variants of the language models, starting by making explicit the multitude of models in a collection. A collection of textual documents can be generated as follows (see Figure 5.8):

For each of the M documents in the collection,

1. pick the corresponding language model $\boldsymbol{\phi}$.
2. For each term in the document,

   draw a random term from $\boldsymbol{\phi}$ according to the multinomial distribution mult($\boldsymbol{\phi}$).

Like in the Gaussian mixture models, a single point estimate of the model that has generated the samples (terms) is used; each document has its own generative model $\boldsymbol{\phi}$. This is similar to the model that Nigam et al. (2000) use

[6] $\mathrm{P}(c|\boldsymbol{\theta})$ is multinomial; components are drawn from $\boldsymbol{\theta}$ with replacement.

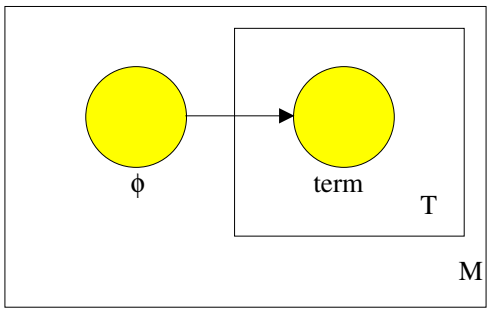


Figure 5.8: Graphical representation of language model for generating collections.

for text classification. They use a mixture of unigrams model, with a one-to-one correspondence between mixture components and classes, thus each class is generated from a separate distribution. While Nigam et al. worked with a relatively small number of classes, in this thesis each documents is a different class.

A Bayesian extension of the language model can be derived by picking a random language model for each document instead of assuming that each document has an observed model. The resulting model is visualised in Figure 5.9. For details on Bayesian language models for information retrieval, the reader is referred to (Zaragoza et al., 2003). Section 5.4.2 discusses a variant that goes beyond single point estimates, without considering all possible generating models. Instead a weighted average over a number of models is taken.

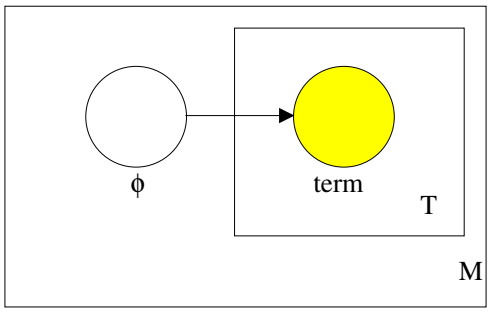


Figure 5.9: Graphical Representation of Bayesian language model.

5.4.2 A pseudo relevance feedback view

Instead of integrating over all possible models, this section assumes we have a set of models that was trained on a reference collection. We concentrate on

Gaussian mixture models, noting that the same techniques can be applied to language models.

Assuming a set of trained Gaussian mixture models $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ automatically re-introduces the restrictions on component selection: each model has its own set of components. Now, in analogy to the Bayesian approach, a natural way to calculate the joint likelihood of query and document samples is to sum over all models in $\boldsymbol{\Theta}$.

$$\mathrm{P}(Q|D,r) \equiv \mathrm{p}(\mathcal{V}_q|\mathcal{V}_d) = \sum_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d) \tag{5.27}$$

$$= \sum_{\boldsymbol{\theta}} \frac{\mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta})\mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{V}_d)}. \tag{5.28}$$

Thus, like in the full Bayesian approach, we consider multiple models and weight their contributions. However, instead of considering all possible models, we only look at a fixed set of models. A pseudo relevance feedback view provides an insightful way of looking at this. First, the document samples are used to find the most likely models $\mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d)$, thus producing a ranking.[7] Then from these ranked models, queries are sampled. Query samples can be generated by any model in the ranking $\mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta})$, but the contribution of a model is weighted by its score in the ranking $\mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d)$. This is similar to pseudo relevance feedback based query expansion, where only the top ranked documents contribute new query terms.

A disadvantage of this approach over the maximum likelihood variant is that it requires to compute $\mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d)$, for each document d in the document collection and for each model $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$. In fact, this means, each document has to be run as a query against the model collection. Fortunately, these probabilities can be pre-computed offline. An important advantage is that the amount of on-line computation needed can be reduced by using a collection of models that is significantly smaller than the collection of documents.

5.5 Multimodal variants

In all graphical models presented in the previous section, except for the full Bayesian ones, each model ($\boldsymbol{\theta}$ or $\boldsymbol{\phi}$) is estimated from a single document. The documents provide a coupling between textual and visual models that

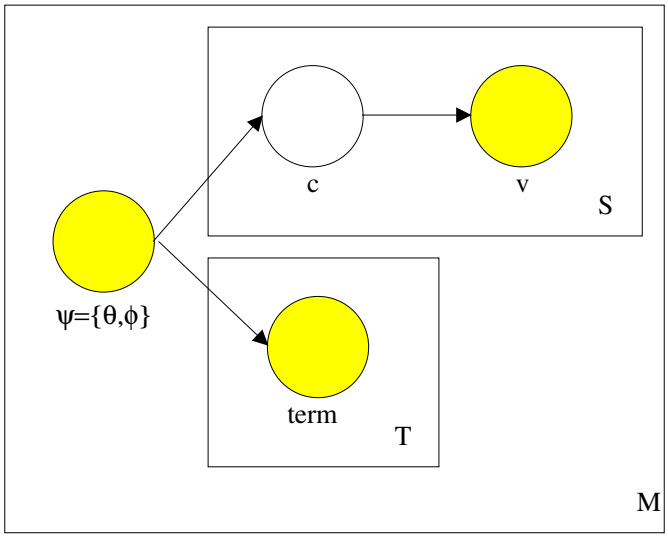
[7] To compute the likelihood of the models, the prior distribution $\mathrm{P}(\boldsymbol{\theta})$ is needed. In absence of information we can assume a uniform prior. $\mathrm{p}(\mathcal{V}_d)$ can safely be ignored, since it does not influence the ranking.

allows for cross-modal tasks like cross-modal retrieval and automatic annotation. In cross-modal retrieval, textual queries can be used to retrieve visual documents (or vice versa). Automatic annotation provides keywords for unlabelled images.

As a first step, we couple the most basic variants of the Gaussian mixture model and the language model. We represent the models for a document d as $\boldsymbol{\psi_d} = \{\boldsymbol{\theta}_d, \boldsymbol{\phi}_d\}$. The procedure for generating a multimodal collection is the following (see also Figure 5.10)

For each of the M documents in the collection,

1. pick the corresponding models $\boldsymbol{\psi_d} = \{\boldsymbol{\theta}_d, \boldsymbol{\phi}_d\}$.
2. For each sample $\boldsymbol{v}$ in the document,
   (a) pick a random component c_i from Gaussian mixture model $\boldsymbol{\theta}_d$ according to the prior distribution over components $\mathrm{P}(c)$ and
   (b) draw a random sample from c_i according to the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$.
3. For each term in the document,
   draw a random term from $\boldsymbol{\phi}_d$ according to the multinomial distribution $\mathrm{mult}(\boldsymbol{\phi}_d)$.



GmmLmAll

Figure 5.10: Graphical representation of generative model for visual samples and textual terms.

In fact, this model is used in Section 4.4.2 for combining textual and visual runs, where $\boldsymbol{\theta}_d$ and $\boldsymbol{\phi}_d$ are the maximum likelihood estimates of the visual samples and textual terms in document d. The joint likelihood of observing query samples and query terms from these models has been used to rank the documents. This model will be called GMMLMALL, since it tries to find the optimal term(s) to annotate the set of all samples.

Because visual samples and textual terms in a document are linked with a single common model Ψ, observing visual samples influences the likelihood for the textual samples. The reason for this is that some models are more likely to have generated the visual samples. Therefore, the probability of observing textual terms from these models will rise. However, as soon as the source of the visual samples is known, information about the visual samples becomes useless for predicting textual terms. Once the model is known the textual terms only depend on that model. So, to infer cross-modal relations, we need to regard the document models $\boldsymbol{\Psi}$ as unobserved variables, as represented in Figure 5.11. This model is similar to the Gaussian Multinomial

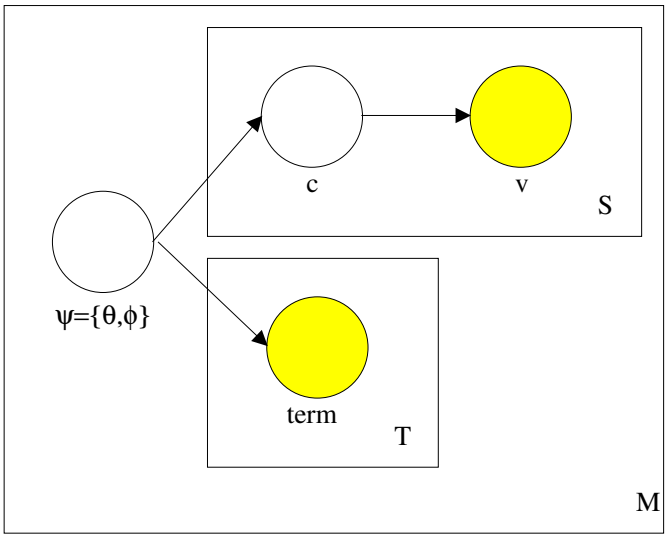


Figure 5.11: Graphical representation of generative model for visual samples and textual terms. Generating models unobserved.

LDA model (GM-LDA) presented by Blei and Jordan for modelling multimodal data (Blei and Jordan, 2003). They use a mixture over aspects for representing the textual part of the modal, similar to the mixture over components for the visual part. We use a single multinomial model to generate textual (annotation) terms. The resulting joint distribution for a multimodal

document $\boldsymbol{d} = (\mathcal{T}, \mathcal{V})$ is:[8]

$$\mathrm{p}(\mathcal{V}, \mathcal{T}) = \int_{\boldsymbol{\psi}} \left(\prod_{\boldsymbol{v} \in \mathcal{V}} \left[\sum_{i=1}^{C} \mathrm{p}(\boldsymbol{v}|c_i) \mathrm{P}(c_i|\boldsymbol{\theta}) \right] \prod_{\mathrm{term}_i \in \mathcal{T}} [\mathrm{P}(\mathrm{term}_{\boldsymbol{i}}|\boldsymbol{\phi})] \, \mathrm{p}(\boldsymbol{\psi}) \right) \mathrm{d}\boldsymbol{\psi} \tag{5.29}$$

Blei and Jordan assume a Dirichlet prior on $\boldsymbol{\psi}$ and use variational inference to estimate the models parameters. As in the previous section, we switch to a reference collection and approximate the joint distribution by summing over a set of reference models $\boldsymbol{\Psi}$

$$\mathrm{p}(\mathcal{V}, \mathcal{T}) = \sum_{\boldsymbol{\psi} \in \boldsymbol{\Psi}} \left[\prod_{\boldsymbol{v} \in \mathcal{V}} \left[\sum_{i=1}^{C} \mathrm{p}(\boldsymbol{v}|c_i) \mathrm{P}(c_i|\boldsymbol{\theta}) \right] \prod_{\mathrm{term}_i \in \mathcal{T}} [\mathrm{P}(\mathrm{term}_{\boldsymbol{i}}|\boldsymbol{\phi})] \, \mathrm{p}(\boldsymbol{\psi}) \right] \tag{5.30}$$

Now, to use this model in a cross-modal setting, one modality can be used to estimate the distribution over the reference collection $\mathrm{p}(\boldsymbol{\psi})$; then this distribution can be used to sample observations in the other modality:

$$\mathrm{p}(\mathcal{V}|\mathcal{T}) = \sum_{\boldsymbol{\psi} \in \boldsymbol{\Psi}} \mathrm{P}(\mathcal{V}|\boldsymbol{\psi}) \mathrm{P}(\boldsymbol{\psi}|\mathcal{T}) \tag{5.31}$$

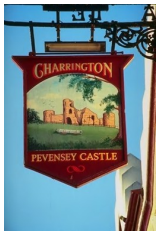
Again, this is a form of pseudo relevance feedback: first, one modality is used to rank the models in the collection; then, the other modality is sampled mainly from the top ranked documents.

Preliminary automatic annotation experiments are conducted using the COREL3892 collection. We estimate a Gaussian mixture model for each image in the collection using the procedure from Section 3.5.1. We also estimate a language models for each image based on the keywords and caption associated with the image (cf. Section 3.5.2).

Four arbitrary images from the collection are used as examples. For each of these the likelihood of the samples given each of the terms in the vocabulary is computed using Equation 5.31. The set of models to marginalise over $\boldsymbol{\Psi}$, is the whole set of models for the COREL3892 collection, except for the model of the image to annotate, thus annotations cannot be inferred from the original keywords. The vocabulary consists of the stemmed terms from all captions. Figure 5.12 shows for each image the 10 terms with highest scores. In practise, often only one model $\boldsymbol{\psi}$ contributes to the results, because the ratio in probability of generating the visual samples ($\mathrm{P}(\mathcal{V}|\boldsymbol{\psi})$) between this top scoring model and the other models is large. Therefore, the contribution

[8] For correspondence between textual and visual parts in the graphical models, we deviate from our standard representation of textual documents as a vector of term counts $\boldsymbol{t}$. Instead, a textual document is represented as a bag of terms $\mathcal{T} = \{\mathrm{term}_1, \ldots, \mathrm{term}_T\}$.





foal	templ	train	**forest**
mare	ruin	emblem	**tiger**
field	sky	**sign**	**cat**
hors	peopl	**sky**	**grass**
in	in	in	in
mose	mose	mose	mose
cradl	cradl	cradl	cradl
the	the	the	the
drop	drop	drop	drop
blackbird	blackbird	blackbird	blackbird

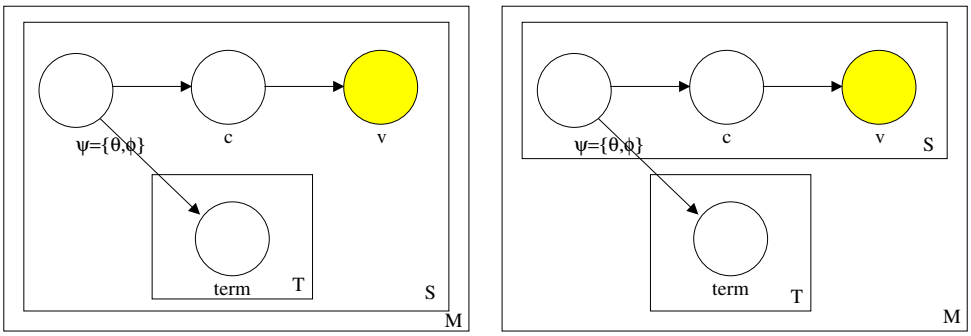
Figure 5.12: Example images with top 10 annotation terms, using the GmmLmAll model from Figure 5.11, equation 5.31 (original keywords in boldface)

of all models except for the most likely one is negligible. This explains why all examples in Figure 5.12 share the lower ranked annotations; the top 4 terms are explained from the model that is most likely to have generated the visual samples, the rest is image independent.

In the following the requirement of a single model that generates all samples $\mathcal{V}$ is dropped. Two variants are discussed: in the first variant (GmmLmEach), we assume observing an annotation term means each of the samples in an image is annotated with this term. The second variant (GmmLmAny), assumes each of the annotation terms is associated with any of the visual samples. Both models are visualised in Figure 5.13. The GmmLmEach model selects for each sample a model $\boldsymbol{\psi}$ and then draws samples $\boldsymbol{v}$ from the components c. In addition, for each sample, T terms are drawn from the multinomial model $\boldsymbol{\phi}$ corresponding with $\boldsymbol{\psi}$. The GmmLmAny model is the same as the Correspondence LDA model (Blei and Jordan, 2003). The procedure is the following:

For each of the M documents in the collection,

1. for each visual sample $\boldsymbol{v}$ in the document,
   (a) pick a random model $\boldsymbol{\psi}$ from the reference collection $\boldsymbol{\Psi}$,



GMMLMEACH GMMLMANY

Figure 5.13: Graphical representation of generative models for visual samples and textual terms. Each sample can come from a different model ψ.

(b) pick a random component c_i from Gaussian mixture model $\boldsymbol{\psi}$ according to the prior distribution over components $\mathrm{P}(c)$ and

(c) draw a random sample from c according to a Gaussian distribution.

2. For each term in the document,

(a) pick one of the samples $\boldsymbol{v}$ from the visual model and

(b) draw a random term from the model $\boldsymbol{\phi}$ that corresponds to the visual model $\boldsymbol{\theta}$ that generated $\boldsymbol{v}$, according to the multinomial distribution $\mathrm{mult}(\boldsymbol{\phi}_d)$.

For both GMMLMEACH and GMMLMANY, we compute the posterior probability of a single term given a single sample for all terms and all samples:

$$\mathrm{p}(\mathrm{term}|\boldsymbol{v}) = \sum_{\boldsymbol{\psi}\in\boldsymbol{\Psi}} \mathrm{P}(\mathrm{term}|\boldsymbol{\psi})\mathrm{P}(\boldsymbol{\psi}|\boldsymbol{v}) \tag{5.32}$$

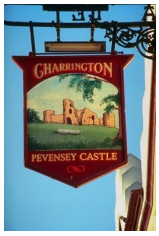
The probability of a term given a set of samples (image) is defined as either the joint probability of all samples (GMMLMEACH), or the marginal over the samples (GMMLMANY).

$$\mathrm{p}_{\textsc{GmmLmEach}}(\mathrm{term}|\mathcal{V}) = \prod_{\boldsymbol{v}\in\mathcal{V}} \mathrm{p}(\mathrm{term}|\boldsymbol{v}) \tag{5.33}$$

$$\mathrm{p}_{\textsc{GmmLmAny}}(\mathrm{term}|\mathcal{V}) = \sum_{\boldsymbol{v}\in\mathcal{V}} \mathrm{p}(\mathrm{term}|\boldsymbol{v})\mathrm{p}(\boldsymbol{v}) \tag{5.34}$$

Figures 5.14 and 5.15 show example annotations found using these models. For each test image, the reference collection $\boldsymbol{\Psi}$ is composed of the models from the COREL3892 test set excluding the test image and another 10% from





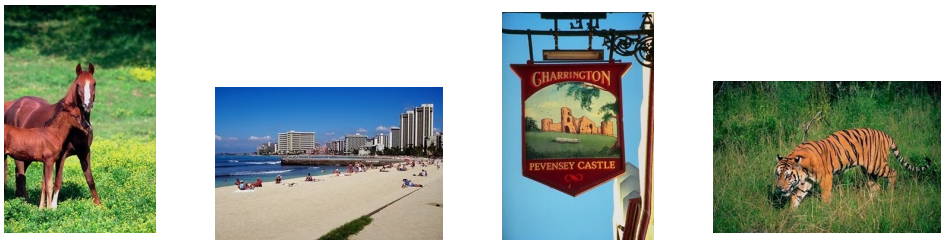
hors	**water**	peopl	**grass**
field	sky	**sky**	tree
peopl	peopl	tree	water
tree	tree	water	hors
water	**sand**	**sign**	peopl
grass	mountain	grass	**cat**
foal	grass	emblem	field
mare	**build**	**build**	sky
sky	rock	hors	**tiger**
flower	**beach**	rock	mare

Figure 5.14: Example images with top 10 annotation terms from GMMLMEACH model (Figure 5.13, Equation 5.33 (original keywords in boldface)

the same class. This roughly corresponds to the 90/10 train/test split used on a comparable taks and dataset by (Duygulu et al., 2002), (Jeon et al., 2003) and (Lavrenko et al., 2004). Both GMMLMANY and GMMLMEACH seem to provide better annotations than GMMLMALL, but of course one cannot draw conclusions from four test images. An extensive evaluation of automatic annotation methods is beyond the scope of this thesis. The examples in this section are only meant to illustrate the possible uses of the models in a cross-modal setting.

5.6 Optimisation

Image retrieval using Gaussian mixture models is computationally expensive. Even in the most basic variant, introduced in Chapter 3, the likelihood $p(\boldsymbol{v_q}|\boldsymbol{\theta_d})$ of each query image sample $\boldsymbol{v_q}$ is computed for each document model $\boldsymbol{\theta_d}$ in the collection. For a typical query image (1,300-1,500 samples), computing this likelihood for all models in the TRECVID2003 collection (32,318 documents), this takes about 15 minutes on a PC with a 1.4GHz Athlon processor and 1GB of memory. The variants presented in this Chap-



hors	sky	**sky**	**grass**
field	**water**	peopl	tree
foal	peopl	tree	hors
mare	tree	**sign**	water
peopl	mountain	water	**cat**
grass	**sand**	emblem	mare
tree	**beach**	grass	peopl
water	snow	**build**	field
flower	grass	hors	foal
sky	**build**	mountain	**tiger**

Figure 5.15: Example images with top 10 annotation terms from GMMLMANY model (Figure 5.13, Equation 5.34 (original keywords in boldface)

ter are at least as expensive. Although efficiency is not the main focus in this thesis, we are aware that for practical usefulness of the methods efficient processing is necessary. As a first study in this direction, two approximations are considered. Section 5.6.1 approximates the probability of a query image given a document model, by the probability of a random subset of the query samples. Section 5.6.2 investigates the comparison of query and document models instead of query samples and document models. Optimising processing without approximation is discussed in (Cornacchia et al., 2004).

5.6.1 Using subsets of query samples

A very basic optimisation strategy is to compute the likelihood for only a subset of the query samples. The intuition behind this is that a large enough subset contains enough information about the example image to produce a reasonable ranking. Adding more samples from the same example, would only add duplicate information, without altering the ranking too much. Obviously, taking fewer samples from an example will speed up the computation as the complexity is linear in the number of samples. First, we consider taking a subset of samples directly from the query image(s). Then, we look at drawing random samples from query models.

Sampling from images

To test the influence of the size of the set of query samples on retrieval effectiveness, the TRECVID2003 collection is used. For each visual example, random subsets of its samples are selected. Each of these sets is used as a query against the TRECVID2003 collection. The size of the sets is varied from 10 to 100 in steps of 10 samples and from 100 to 1,000 in steps of 100 samples Also the full set of samples is used as a query (i.e., the whole example image) Samples are drawn uniformly from the full set of samples, without replacement, so no duplicate samples are selected. For each setting the sampling is repeated 10 times to reduce chance effects. We will call this variant QSAMPLEUNIFORM.

In a second sampling variant, QSAMPLEPOSITION, instead of drawing samples uniformly, we used position information to select a subset of samples. The idea is that it is important to carefully select the samples, especially if only a small set of samples from the query image is used. Typically, the important information in an image resides roughly in the middle of the image plane, thus it seems wise to take more of those samples. Nevertheless, we do not want to completely ignore the information outside of the centre. Therefore, a positional mixture distribution over the image plane is defined

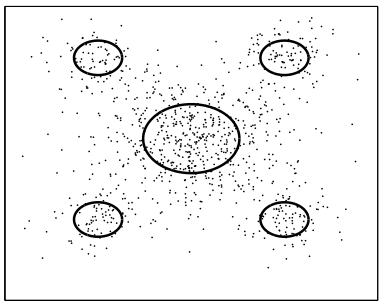


Figure 5.16: Positional mixture (unit variance contours) and random sample from it (points)

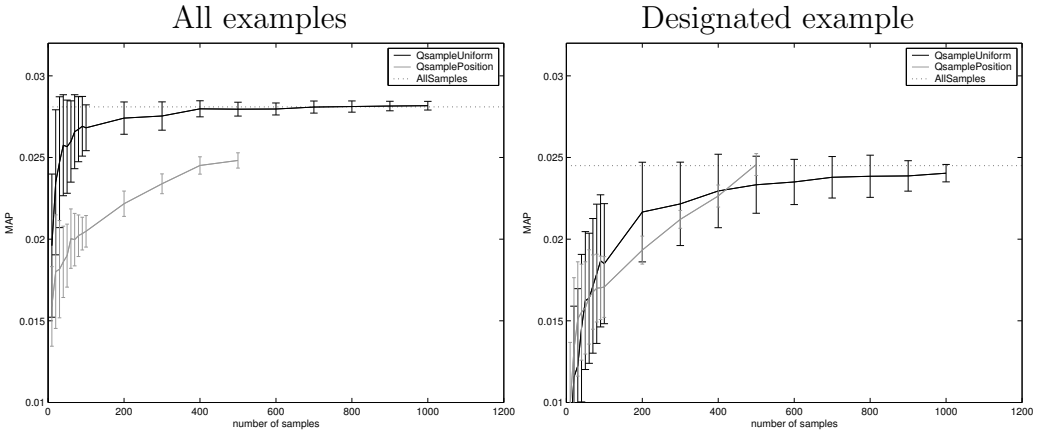


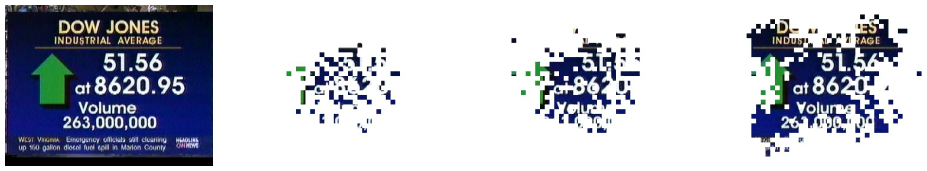
Figure 5.17: MAP for using different numbers of samples from the query images: all query examples (left) and designated examples only (right).

that mainly generates central positions, but also positions in the corners. Figure 5.16 shows the unit variance contours of the positional mixtures five components and a 1,000 point random sample from the distribution. It gives an idea of the preferred positions of query samples for the QSAMPLEPOSITION variant. Like in the first variant, no duplicate samples are selected. In this variant the maximum number of samples used is 500, because we sample without replacement, using more would be close to a uniform sampling strategy.[9]

Figure 5.17 shows the results obtained for the two sampling variants, for a designated examples run and an all examples run. The average MAP over the 10 experiments is shown along with the variance.

Obviously, as sample sizes get larger, the results are less dependent on

[9]Also sampling randomly from the position mixture until enough unique values are found can take long.



example image 100 samples 200 samples 500 samples

Figure 5.18: Designated example for Dow Jones topic (VT0120), with random samples drawn using QSAMPLEPOSITION.

the actual sample and variance gets smaller. Randomly drawing around 400 samples uniformly from the query examples, already gives results that are as good as using all examples. Surprisingly, in the all samples run position based sampling seems a bad idea. Apparently, in many examples the important samples are not in the densely sampled regions. In the designated examples however, this seems to be the case. In fact, using only 500 samples in the QSAMPLEPOSITION yields better results than using all examples. An explanation could be that the user who selected the designated examples preferred examples that showed the object(s) of interest in the centre of the image plane. However, looking at the individual topic scores, it is apparent that the effect can be mainly attributed to a single topic requesting Dow Jones graphics (VT0120), see Figure 5.18. Apparently, the central white lettering on a dark background is enough to retrieve the relevant graphics.

The plots of the MAP scores hide some information. They only show the effects of sampling on relevant documents and thus emphasise topics with high scores (like the Dow Jones topic). To look beyond relevant documents, we compute the correlation between the sampling variants and the original ranking, using Kendall's correlation coefficient, known as Kendall's τ (see for example Conovar, 1980). Plots of this coefficient against the sample size are shown in Figure 5.19. The designated example plots shows that position based sampling is unrelated to the original ranking and thus perhaps is a bad idea. For the other sampling variants, the plots show that, as expected, the correlation goes up as the number of samples increases.

Sampling from models

Instead of taking samples directly from query images, it is interesting to compute the likelihood for samples that are drawn from query models. The query models are abstractions of the query example(s). A random sample from a query model is an approximation of the query and can be used to rank the document models in the collection. To test whether this approximation is useful for retrieval, we take the full example image models (see Section 5.2)

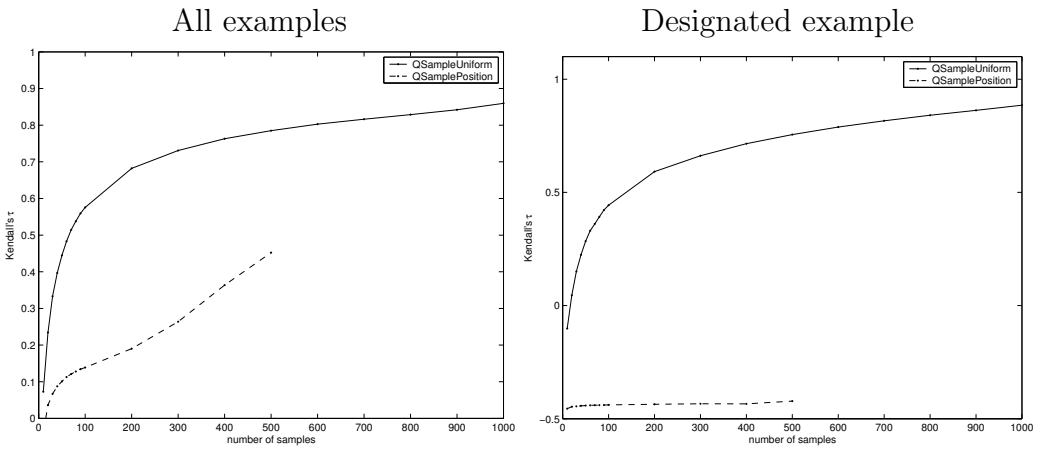


Figure 5.19: Kendall's τ between different sampling variants and using full query.

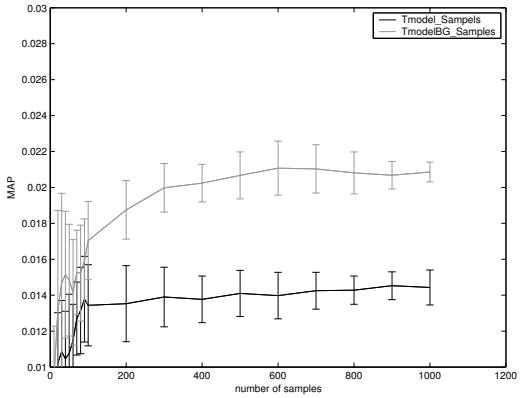


Figure 5.20: MAP for using different numbers of samples from query models.

and draw random samples of different sizes from them (between 10 and 1,000 samples). For each set of samples, the document models in the collection are ranked based on their likelihood of generating the samples and mean average precision is calculated. To reduce chance effects, the process is repeated 10 times for each sample size and the mean and standard deviation scores over the 10 trials for the different sample set sizes are computed. The experiment is repeated for query models that are built using background probabilities during training (see Section 5.3). Figure 5.20 shows the results.

Clearly the mean average precision scores for samples from query models are much lower than the scores for samples from query images (cf. Figure 5.17). An explanation could be that we have introduced too much uncertainty. Both query and document are abstract representations of the original images. This makes sense in retrieval, since to go beyond exact matching,

it is necessary to abstract away from the exact representation. When either query or document is represented by a probabilistic model, some variation in the exact representation is allowed. However, it seems modelling both query and document probabilistically introduces too much variation.

5.6.2 Asymptotic likelihood approximation

Instead of randomly sampling from the query model and then computing the likelihood of these generated samples given the document model, it is possible to compare the two models directly using the Kullback-Leibler-divergence (KL). The KL-divergence, also known as relative entropy, measures the amount of information there is to discriminate one model from another. For Gaussian mixture models no closed-from solution exists for this KL-divergence. However, under the assumption that the Gaussians are well separated, Vasconcelos derives an approximation: the Asymptotic Likelihood Approximation (ALA) (Vasconcelos, 2000).

$$\begin{aligned}
&\mathrm{KL}[\mathrm{p}(\cdot|\boldsymbol{\theta_q})||\mathrm{p}(\cdot|\boldsymbol{\theta_d})] \approx \\
&\mathrm{ALA}[\mathrm{p}(\cdot|\boldsymbol{\theta_q})||\mathrm{p}(\cdot|\boldsymbol{\theta_d})] = \sum_{c=1}^{N_C} \mathrm{P}(c_{q,c})\{\log\ \mathrm{P}(c_{d,\alpha(c)}) \\
&\qquad + \log\ \mathrm{p}(\boldsymbol{\mu_{q,c}}|\mathcal{N}(\boldsymbol{\mu_{d,\alpha(c)}}, \boldsymbol{\Sigma_{d,\alpha(c)}})) \\
&\qquad - \frac{1}{2}\mathrm{trace}[\boldsymbol{\Sigma^{-1}_{d,\alpha(c)}}\boldsymbol{\Sigma_{q,c}}]\}, \\
&\text{where } \alpha(c) = k \Leftrightarrow \|\boldsymbol{\mu_{q,c}} - \boldsymbol{\mu_{d,k}}\|_{\boldsymbol{\Sigma_{d,k}}} \leq \|\boldsymbol{\mu_{q,c}} - \boldsymbol{\mu_{d,l}}\|_{\boldsymbol{\Sigma_{d,l}}} \forall l \neq k
\end{aligned} \tag{5.35}$$

In this equation, $\mathrm{p}(\cdot|\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}))$ is the probability that a sample is drawn from the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Furthermore, subscripts indicate documents and components (e.g. $\boldsymbol{\mu_{d,c}}$ is the mean for component c_c of document model $\boldsymbol{\theta_d}$).

Experiments

To compare the asymptotic likelihood assumption measure to the sample likelihood or query likelihood used in the previous chapters, we repeated some of the experiments from Chapter 4 using ALA as the measure. Therefore, in addition to models for all documents in the collection, models for the queries are needed. The experiments reported here use the COREL3892 and TRECVID2002 test collections. Gaussian mixture models are built for each individual visual example in these collections. In addition, for TRECVID2002, topic models are built from all available examples for each

Table 5.3: MAP scores for asymptotic likelihood assumption (ALA) and query likelihood (QL) for different collections and query types.

Collection	Query Type	ALA	QL
COREL3892	single	.079	.123
TRECVID2002	all	.006	.029
TRECVID2002	best	.014	.044
TRECVID2002	designated	.008	.028

topic, i.e., for each topic, a single model was trained to represent all examples for the topic. This way we can compare ALA to sample likelihood for all, best and designated example queries (see Section 4.3.1). For COREL3892, only single example queries are used. The ALA between each topic model and each collection model is computed and the resulting scores are used to rank the documents in the collection. Table 5.3 shows the results along with the original query likelihood (QL) results. In all cases, QL is significantly better than ALA. This can possibly be explained by the amount of variability introduced by representing both query and document probabilistically. The same effects are noticed with the random samples from the query models (Section 5.6.1). Another explanation could be that the ALA assumptions are too strong and consequently ALA results are not trustworthy. The next section takes a closer look at these assumptions.

ALA assumptions

The main assumption behind the ALA is that the Gaussians for the components c_c within a class model $\boldsymbol{\theta_d}$ have small overlap. In fact, there are two parts to this. The first assumption is that each image sample is assigned to one and only one of the mixture components. The second is that samples from the support set of a single query component are all assigned to the same document component (Vasconcelos, 2000). More formally:

Assumption 5.6.1 *For each sample, the maximum posterior probability for a component equals 1:*

$$\forall \boldsymbol{\theta_d}, \boldsymbol{x} : \max_k P(c_{d,k}|\boldsymbol{x}) = 1$$

Assumption 5.6.2 *For any document model $\boldsymbol{\theta_d}$, the component with maximum posterior probability is the same for all samples in the support set of a single query component $c_{q,k}$:*

$$\forall c_{q,k}, \boldsymbol{\theta_d}\ \exists l^*\ \forall \boldsymbol{x} : p(\boldsymbol{x}|c_{q,k}) > 0 \implies \arg\max_l\ P(c_{d,l}|\boldsymbol{x}) = l^*$$

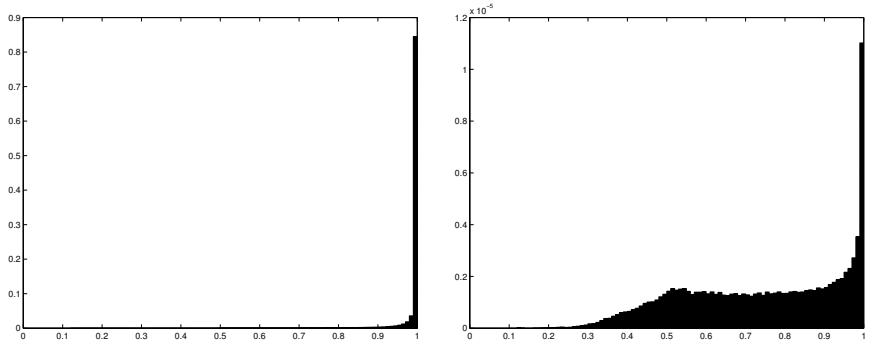


Figure 5.21: Testing the ALA assumptions A (left) and B (right). The samples $\boldsymbol{x}$ drawn from $\mathrm{p}(\boldsymbol{x}|c_{d,k})$. Histograms of the maximum posterior component assignments within document model $\boldsymbol{\theta_d}$ and in different (randomly chosen) models $\boldsymbol{\theta_{d'}}$ are shown.

Assumptions 5.6.1 and 5.6.2 have been tested as follows, using Monte Carlo simulation (Metropolis and Ulam, 1949). First, a random document model $\boldsymbol{\theta_d}$ is taken from the search collection and then a random mixture component $c_{d,k}$ from the mixture model of this document is selected. Then, 10,000 random samples are drawn from this component and for each sample $\boldsymbol{x}$ the following measures are computed:

- $\mathrm{P}(c_{d,l}|\boldsymbol{x})$, the posterior component assignment for all components $c_{d,l}$ within document model $\boldsymbol{\theta_d}$.
- $\mathrm{P}(c_{d',m}|x)$, the posterior component assignment for all components $c_{d',m}$ in a different randomly chosen document model $\boldsymbol{\theta_{d'}}$.

For the first measure the maximum posterior probability for each sample is stored. The second measure is averaged over all 10,000 samples to obtain the proportion of samples assigned to each component. The maximum over all components is then taken to find the proportion of samples assigned to the most probable component (remember, there should be a component that explains all samples). The whole process is repeated 100,000 times for different documents and components selected at random. The results are histogrammed in Figure 5.21. Both measures should be close to 1, the first to satisfy Assumption 5.6.1, the second to satisfy assumption 5.6.2. As is clear from the plots, Assumption 5.6.1 appears reasonable, but Assumption 5.6.2 is too strong.[10] This may also explain the difference between ALA results and QL results in the previous section. Goldberger et al. (2003) propose two

[10]Note that the scales on the y-axes for the two plots differ.

different approximation of the KL-divergence between Gaussian mixtures. The first is also based on Assumption 5.6.2 and differs from ALA only in the way of computing the most similar component. The second variant is similar to Monte Carlo methods (Metropolis and Ulam, 1949), but instead of sampling randomly from the query model, Goldberger et al. use points on the unit covariance hypersphere. This way, they avoid taking Assumption 5.6.2, at the cost of some extra computation.

5.7 Summary

This Chapter investigated alternative ways of using the generative models introduced in Chapter 3. All approaches presented here can be used within the generative probabilistic retrieval framework presented in Section 5.1. The approaches differ in the choice of factoring the conditional probability $\mathrm{P}(d, q|r)$ in a query or document generation fashion and in the estimation of the probabilities $\mathrm{P}(q|d, r)$, $\mathrm{P}(d|q, r)$ and $\mathrm{P}(d|\bar{r})$. Figure 5.22 summarises the choices for the different variants. For simplicity, smoothing is ignored.

The models built from the query samples in the document generation approach can be seen as relevance models (cf. Lavrenko and Croft, 2003) and they can be updated once additional relevance information is available. This facilitates relevance feedback, which is problematic in the query generation approach, where the knowledge that document A is relevant does not change the model for B (see Section 3.7).

Some of the approaches discussed in this chapter need a set of reference models $\boldsymbol{\Theta}$ either to compute joint query and document probabilities, or to compute background probabilities. So far, we have simply used a sample from a highly comparable collection (the TRECVID2003 development set). Alternatively, it is possible to automatically find an optimal set of models to describe a collection using the LDA approach (Section 5.4.1). In LDA a variational inference procedure is used to simultaneously find the distribution over latent topic models and the sample distributions for each topic that together best describe a training collection (Blei et al., 2003). Since these distributions together are a good description of the collection statistics, they can be used as a background model.

Query Generation, Chapter 3

$\textsc{rsv}(d) = \mathrm{P}(q|d,r) \equiv \mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta}_d)$

Document Generation, Section 5.2

$\textsc{rsv}(d) = \frac{\mathrm{P}(d|q,r)}{\mathrm{P}(D|\bar{r})} \equiv \frac{\mathrm{p}(\mathcal{V}_d|\boldsymbol{\theta}_q)}{\mathrm{p}(\mathcal{V}_d)}$

Background training, Section 5.3

Same as either query or document generation variant, but background probabilities are used in estimation of the models $(\boldsymbol{\theta}_q, \boldsymbol{\theta}_d)$.

Bayesian approach, Section 5.4

$\textsc{rsv}(d) = \mathrm{P}(q|d,r) \equiv \mathrm{p}(\mathcal{V}_q|\mathcal{V}_d) = \int_{\boldsymbol{\theta}} \mathrm{p}(\mathcal{V}_q|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}|\mathcal{V}_d)\mathrm{d}\boldsymbol{\theta}$

Multi-modal, Section 5.5

For example annotation, visual query to find textual terms (single term documents):
$\textsc{rsv}(d) \equiv \mathrm{p}(\mathcal{V}_q|\mathcal{T}_d) = \sum_{\boldsymbol{\psi}\in\boldsymbol{\Psi}} \mathrm{P}(\mathcal{V}_q|\boldsymbol{\psi})\mathrm{P}(\boldsymbol{\psi}|\mathcal{T}_d)$,
where $\boldsymbol{\Psi}$ is a reference collection.

Query subsets, Section 5.6.1

$\textsc{rsv}(d) = \mathrm{P}(q|d,r) \equiv \mathrm{p}(\mathcal{V}_q^{\text{sample}}|\boldsymbol{\theta}_d)$,
where $\mathcal{V}_q^{\text{sample}} \subset \mathcal{V}_q$ (sampling from images),
or $\mathcal{V}_q^{\text{sample}}$ is drawn randomly from $\mathrm{p}(\cdot|\boldsymbol{\theta_q})$ (sampling from models).

ALA, Section 5.6.2

$\textsc{rsv}(d) = \mathrm{P}(q|d,r) \equiv \mathrm{KL}[\mathrm{p}(\cdot|\boldsymbol{\theta_q})||\mathrm{p}(\cdot|\boldsymbol{\theta_d})] \approx \mathrm{ALA}[\mathrm{p}(\cdot|\boldsymbol{\theta_q})||\mathrm{p}(\cdot|\boldsymbol{\theta_d})]$

Figure 5.22: Summary of approaches.

6

Evaluating multimedia retrieval

This chapter discusses how multimedia retrieval techniques can be evaluated. Systematic evaluation of multimedia retrieval has gained attention only recently. The chapter starts with a short history of the evaluation of multimedia retrieval approaches in general, and content-based approaches in particular. Then, starting from the well-established text retrieval evaluation methodology, the chapter discusses how multimedia retrieval test collections can be build. We argue that laboratory testing, as used in text retrieval, is a useful technique for evaluating multimedia retrieval approaches. Section 6.2 introduces the laboratory tests that are now common in text retrieval. Section 6.3 discusses how to build a test collection for evaluating multimedia retrieval approaches. Section 6.4 introduces techniques for judging multimedia retrieval performance. Section 6.5 summarises the chapter, and reviews the evaluation methodology applied in the previous chapters. Parts of the discussion in this chapter have been published in (Westerveld and De Vries, 2003b), and (Westerveld and De Vries, 2003a).

6.1 History

Until recently no commonly used evaluation methodology existed for content-based image and video retrieval. An important reason for this is that for long, the field has been merely a showcase for computer vision techniques. Many papers in the field 'proved' the technical merits and usefulness of their approaches to image processing by showing a few well-chosen, and well-performing examples. Since 1996 the problem of systematically evaluating multimedia retrieval techniques has gained more and more interest. In that year, the MIRA (Multimedia Information Retrieval Applications) working group was formed (Draper et al., 1999; Dunlop, 2000). The group, consist-

ing of people from the fields of information retrieval, digital libraries, and library science, studied user behaviour and information needs in multimedia retrieval situations. Based on their findings, they developed performance measures. Around the same time, in the multimedia community the discussion on proper evaluation started, and Narasimhalu et al. (1997) proposed measures for evaluating content-based information retrieval systems. These measures are based on comparing ranked lists of documents returned by a system to the perfect, or ideal, ranking. However, they do not specify how to obtain such a perfect ranking, nor do they propose a common test set. A year later, Smith (1998) proposed to look at the text retrieval community, and to use measures from TREC for image retrieval evaluation. Again, no dataset was proposed. At the start of the 21st century, the evaluation problem gained more attention within the content-based image retrieval community, with the publication of three papers discussing benchmarking in visual retrieval (Müller et al., 2001; Leung and Ip, 2000; Gunther and Beretta, 2000). These three papers call for a common test collection and evaluation methodology and a broader discussion on the topic. The BENCHATHLON network[1] was started to discuss the development of a benchmark for image retrieval. Then, in 2001, TREC started a video track (Smeaton et al., 2002b,a) that evolved into the workshop now known as TRECVID (Smeaton and Over, 2003; Smeaton et al., 2003a).

6.2 Laboratory tests in information retrieval

Information retrieval is interactive. In web search, for example, queries are often changed or refined after an initial set of documents has been retrieved. In multimedia retrieval, where browsing is common, interactivity is perhaps even more important. Saracevic (1995), and Sparck Jones and Willett (1997a) argue that evaluation should take interactivity into account, and measure user satisfaction. Tague-Sutcliffe (1992) called evaluation of a system as a whole in an interactive setting an *operational test*. Such tests measure performance in a realistic situation. Designing such an operational test is difficult and expensive: many users are needed to free the experiment of individual user effects, the experimental setup should not interfere with the user's natural behaviour, and learning effects need to be minimised. Also, because there are many free variables, it is hard to attribute observations to particular causes. In contrast to these tests in fully operational environments, Tague-Sutcliffe defined *laboratory tests* as those tests in which

[1] `http://www.benchathlon.net`

possible sources of variability are controlled. Thus, laboratory tests can provide more specific information, even though they are further away from a realistic setting. Also, laboratory tests are cheaper to set up, because the interactive nature is ignored, and the user is removed from the loop. Laboratory tests measure the quality of the document ranking instead of user satisfaction.

Within the laboratory setting, we distinguish two types of tests: *system-oriented* tests, and *task-oriented* tests. System-oriented tests measure if the system functions properly. Task-oriented tests measure how useful that functionality is for a given task. For example, content-based information retrieval methods claim to identify visually similar documents. A system-oriented test then evaluates whether the retrieved documents are indeed visually similar. A task-oriented test evaluates how useful this is to satisfy an information need in a multimedia retrieval setting. The laboratory style setup for both types of tests is the same. The tests differ in the choice of collection and the way of judging relevance. The next section introduces the laboratory setup commonly used in current text retrieval evaluations (e.g., Voorhees and Buckland, 2004, 2003).

6.2.1 The Cranfield tradition

As stated before, text retrieval has a long tradition of experimentation. Most current evaluation procedures, including TREC, are laboratory tests, based on the CRANFIELD paradigm (Cleverdon, 1967). This section provides a short introduction to this paradigm. A thorough review of the fundamental assumptions behind CRANFIELD style experiments can be found in (Voorhees, 2002).

The term laboratory tests will be used to refer to tests following this paradigm. A test collection for laboratory tests consists of a fixed set of documents, a fixed set of topics, and a fixed set of relevance judgements. Documents are the basic elements to retrieve, topics are descriptions of the information needs, and relevance judgements list the set of relevant documents for each topic. The focus in laboratory tests is on comparative evaluation. Different approaches are tested, and their relative performance is measured. The process is as follows. Each approach produces a ranked lists of documents for each topic. The quality of the ranked lists is measured based on the positions of the relevant documents in the list. The results are averaged across all topics to obtain an overall quality measure.

A number of aspects influence the reliability of evaluation results. First, a sufficiently large set of topics is needed. Sparck Jones and Van Rijsbergen (1976) suggest a minimum of 75. Second, the measures should be stable.

This means it should not be influenced too much by chance effects. Clearly, measures based on few observations are less stable than measures based on many observations. For example, precision at rank 1 (is the first retrieved document relevant) is not a very stable measure. Third, there needs to be a reasonable difference between two approaches before deciding one approach is better than the other. Sparck Jones (1974) suggests a 5% difference is noticable, and a difference greater than 10% is material. Finally, the relevance judgements on which all measures are based should be reliable. The following sections discuss the details of these four conditions.

6.2.2 Reliable measures

The first three conditions are clearly interrelated. For example, when stable measures are used, fewer topics are needed; and when many topics are used, a smaller difference in scores can lead to the conclusion that two approaches are different. Buckley and Voorhees (2000) have investigated these three conditions. They used results from the query track (Buckley and Walz, 2000), which contains 21 different formulations for each of 50 topics. This allowed them to do multiple experiments using the same set of topics and relevance judgements. In each experiment, they compared two approaches based on one of the studied measures. Based on this measure, they concluded whether approach A was better than approach B, whether B was better than A, or whether A and B had the same score (within some predefined margin). The experiment was repeated for 36 pairs of approaches. Buckley and Voorhees then computed the error rate of the measure under study as the fraction of cases that led to a conclusion different from the majority conclusion. The whole experiment was repeated 50 times with different topic formulation sets, obtained by permuting the formulations for a topic over the sets. Thus, they obtained average error rates for each of the measures studied. The experiments were carried out for different measures, different topic set sizes, and different margins for score similarity. Buckley and Voorhees conclude that some measures are more stable than others (i.e., have smaller error rate). Especially measures that are computed on only a small fraction of the ranked lists are instable (e.g., precision@N, with $N \leq 30$). Buckley and Voorhees note however that this does not necessarily mean these measures are useless. A measure should be chosen with respect to a task, and it should measure the aspect of interest for that task. For web retrieval, for example, precision@20 is a reasonable measure, because people often only look at the first few pages of results. Because of the instability of this measure however, more topics are needed to obtain reliable results, or a greater difference in scores between two approaches is needed to conclude they are different. In general, Buckley

and Voorhees recommend using at least 25 topics, but preferably 50. For unstable measures, more topics are needed.

6.2.3 Reliable judgements

For investigating the final condition for reliable evaluation results, the reliability of the relevance judgements, we need to look at the assumptions made with regard to relevance in laboratory tests. The main assumptions are the following. First, relevance is approximated by topical similarity: a document is relevant if it is on topic, i.e., if it discusses the topic of the query. This means the information need is assumed not to change over time. It also means relevance is judged independently for each document. If a document contains information that is on topic, but all this information is already present in other documents, the document is still regarded relevant. The second assumption is that relevance judgements are representative of a user population. Although the judgements are a single person's opinion, they are assumed to be representative of the typical user. Third, judgements are assumed to be complete. For each topic, all relevant documents in the collection are identified. Finally, judgements are often assumed to be binary, i.e., a document is either relevant to a topic or it is not. The original CRANFIELD experiments used graded relevance judgements on a five-point scale, but most modern laboratory tests assume binary judgements.

Clearly, these assumptions do not hold. Relevance judgements from a single user do not represent the opinion of a whole population, topical similarity is not the same as utility, and in many cases it is impossible to identify all relevant documents in a collection. However, the goal in laboratory tests is to compare retrieval strategies, not to find an indication of their absolute performance. Therefore, even though the assumptions may not be strictly true, laboratory tests may be useful. The concern is not so much about the truth in the assumptions, but about the influence of the assumptions on relative scores. A number of studies have investigated how violation of the assumptions influences comparative results (Zobel, 1998; Voorhees, 2002; Voorhees and Harman, 2000). The findings of these studies are highlighted below.

Incomplete judgements

A typical TREC collection consists of between 500,000 and 1,000,000 documents, and is thus far too large to obtain complete relevance judgements. Instead, current laboratory tests use a pooling technique (Sparck Jones and Van Rijsbergen, 1975). Pooling is the process of forming a pool, or set, of the

top ranked documents from a variety of different approaches. Each approach composes a ranked list for each of the topics in the collection. For each topic a pool is constructed, consisting of the union of the top N retrieved documents from all approaches. Only the documents in the pools are judged for relevance. Documents not retrieved within any approach's top N are assumed not relevant. The idea behind this approach is that documents that are not retrieved at a high rank by any system are unlikely to be relevant.

This assumption may not be valid. Indeed both Harman (1996) and Zobel (1998) show some of the unjudged documents are relevant. This could potentially influence the results since systems are usually evaluated on a top $M > N$. However, if the pool is large and diverse enough, that is, if many different techniques contributed to the pool, then the fact that some relevant documents are missing is assumed to be of little consequence. For text retrieval, pool quality has been intensively studied. Zobel (1998) found that the fact that measurements are calculated on a top $M > N$ does not influence comparative results, i.e., the relative ranking of the approaches does not change.

Another concern with incomplete relevance judgements, is their usefulness for evaluating approaches that have not contributed to the pool. The set of relevance judgements could potentially be biased against them. Zobel (1998) investigates this effect by recomputing scores for each approach. The new scores are based on a pool from which the documents that have been uniquely contributed by that approach are removed. He finds this has little effect on the relative ranking of the approaches. Voorhees and Harman (2000) show that even if all approaches from a single group are ignored in constructing the pool, this hardly influences relative results. The exceptions are the interactive runs. If their contributions are removed from the pool, they often end up lower in the ranking of approaches. This means, one has to be careful in evaluating interactive approaches using a fixed test collection, but test collections are valuable resources for evaluating additional automatic methods that did not contribute to the pool. Another danger is that pooling effects are small because many similar approaches contribute to the pool. When the contributions of one approach are removed, the pool contents hardly changes since a similar approach is bound to have found almost the same set of documents. Therefore, fixed test collections may be most useful for evaluating variants of existing techniques.

Subjective judgements

It is well known that relevance judgements are subjective. Different judges will have different opinions on the relevance of documents (e.g., Harter, 1996).

Since the focus is on comparative results this is not necessarily problematic. As Voorhees (1998) states:

> For a test collection, the important question is not so much how well assessors agree with one another, but how evaluation results change with the inevitable differences in assessment.

Voorhees (1998) investigates the influence of difference in assessments on evaluation results by having topics judged by multiple assessors. The different approaches have been evaluated using different combinations of judgements, and ranked by mean average precision. Voorhees finds the resulting rankings are highly correlated, and concludes comparative results are stable with regard to the subjectivity in relevance judgements.

6.3 Multimedia test collections

The previous section summarised the conditions under which laboratory tests in the CRANFIELD tradition are useful tools for evaluating retrieval approaches. The main conditions are: stable measures, sufficient number of topics, and either complete relevance judgements, or a high quality pool. An important condition that is not discussed yet is realism. The documents, topics, and judgements of the test collection should be close to the realistic situation of interest. If the usefulness of an information retrieval approach for searching a collection of medical images is to be tested, clearly the test collection should be of the same type. In this thesis we are interested in disclosing heterogeneous collections, for example for searching media archives of newspapers or broadcasters. This section discusses the construction of a realistic test collections for the evaluation of multimedia retrieval from heterogeneous data sets. The three components of a test collection, documents, topics, and relevance judgements, are treated separately. Each of the Sections 6.3.1, 6.3.2, and 6.3.3 discusses one of them.

6.3.1 Documents

Tague-Sutcliffe (1992) listed the variables one has to consider when constructing a set of documents as part of an information retrieval test collection. These variables include, size, coverage, form and medium. Tague-Sutcliffe did not provide any guidelines other than that a collection has to be *large enough*. She did not indicate when a collection is large enough, but stated that results with a small collection cannot necessarily be extrapolated to a

larger collection. Therefore, the size of the collection should be comparable to the size of a realistic document collection. The same reasoning can be used for the other variables. The set of documents in the test collection should resemble a set of documents in a realistic task. Realistic multimedia collections are huge. For example, a photo archive of a press agency or newspaper contains hundreds of thousands of photos (Sormunen et al., 1999; ANP Beeld, 2004), and the Dutch broadcasting archive contains 70,000 hours of video material (NPS Klokhuis, 2004). For copyright reasons, it is difficult to obtain such large collections, especially if they are to be used by a larger community. Therefore, in a task-oriented evaluation, a balance has to be found between size and availability. System-oriented evaluations ignore the task in which a system is to be used. Instead, they focus on certain aspects of the system. In the remainder of this chapter we assume the aspect of interest in system-oriented laboratory tests is visual similarity. In such a setting, the resemblance of a document collection to a realistic one is less important. The main criteria for the document set in a system-oriented test is that it contains a mix of visually similar and visually dissimilar documents.

Corel and Trecvid are the two most commonly used large document collections for evaluating content-based multimedia retrieval (Section 2.5 describes the collections). These collections are discussed in detail below. Other multimedia collections that are sometimes used in evaluation include the Brodatz and Columbia collections. These collections are especially useful for system-oriented testing. They are designed to capture specific aspects of the information retrieval problem. The Brodatz collection is a set of texture images taken from a photographic album for artists and designers (Brodatz, 1966), and is thus useful for evaluating texture classification algorithms (Picard et al., 1993). The data set is part of the MeasTex image texture database and test suite[2]. The Columbia dataset consists of 100 brightly coloured objects each photographed from many different viewpoints; it is useful for testing colour and shape-based algorithms, and of course viewpoint invariance. Also Clef and Benchathlon have started to set up multimedia test collections. Clef is an evaluation forum for cross-language retrieval focusing on European languages (Peters et al., 2002, 2003). In 2003, it started an image retrieval task (Clough and Sanderson, 2003). The document collection is a subset of 30,000 historic photographs in Scotland from the St. Andrews collection (Reid, 1999), with captions in English. The setting for this ImageClef is similar to the Trecvid setting, but it allows for cross-language retrieval since topics are available in multiple languages. Benchathlon is in the process of collecting data and setting up an image

[2] http://www.cssip.uq.edu.au/meastex/meastex.html

retrieval evaluation framework.[3]

Corel

The Corel document set is a collection of stock photographs, which is divided into subsets each relating to a specific theme (e.g., *tigers*, *sunsets*, or *English pub signs*). A large number of publications uses this image collection to evaluate or illustrate the effectiveness of a given retrieval approach (e.g., Blei and Jordan, 2003; Jeon et al., 2003; Duygulu et al., 2002; Barnard et al., 2003; Belongie et al., 1998; Vasconcelos and Lippman, 2000; Li and Wang, 2003). One problem with the Corel images is that the data is sold commercially, on separate thematic cds. Therefore, obtaining the full collection (over 800 themes, containing over 80,000 images in total) is costly. As a consequence, each group uses their own subset and a single, commonly used Corel subset does not exist. A study by Müller et al. (2002) showed that evaluations using Corel are highly sensitive to the subsets used. This makes direct comparisons of systems on these datasets impossible. In addition, because of its organisation into themes, and the production process where photographs in the same theme are often shot in batch at the same location, the collection is a set of clusters that are far more homogeneous than can be expected in a realistic setting. Consequently, Corel is more suitable for system-oriented evaluation than for a task-oriented test.

Trecvid

The Trecvid document collection is a collection of video data.It grew over the years from 11 hours of data (about 6,300 shots or documents in the test collection) in 2001 to roughly 133 hours (32,000+ shots) in 2003. The number of documents in the Trecvid2003 collection is still nowhere near the size of realistic archives, or of the primary Trec collections for text retrieval evaluation (between 500,000 and 1,000,000 documents). Still, it is the largest available video retrieval evaluation collection to date. The Trecvid2003 collection consists of US broadcast news from 1998. The domain of news broadcasts may be limited, but it is still quite broad, since potentially any topic can be news. A greater problem is that the data come from just a few sources, the main ones being abc and cnn. This introduces the risk of over-fitting. A number of source specific recurring shot types exist in the collection, for example the studio shots of the anchor persons, or the graphics for the weather forecasts. These are probably at least as homogeneous as the themes in the Corel collection and should not be used

[3] http://www.benchathlon.net

- Scenic pictures of lakes and waterways
- Outdoor pictures of an interview with James Cameron discussing the making of the Titanic film
- Pictures of a spooky Eiffel tower
- Images of a sunny day on the Amsterdam Canals
- A funny movie scene
- A photograph of a river in Colorado with trees nearby

Figure 6.1: Some *made-up* information needs from the literature

in a task-oriented evaluation. Another property of the domain, possibly influencing the evaluation results, is the text-oriented nature of news. Shots in a news broadcast are typically accompanied by a voice-over that either describes the visual material or gives some contextual information. This may introduce a bias for text-oriented methods. Indeed, text oriented methods are found to outperform visual ones (Smeaton et al., 2003b) (see also Chapter 4).

6.3.2 Topics

In many papers on visual information retrieval, the authors give examples of queries that users might ask. However, most of the time, these examples do not come from real users, but they are picked out of the blue. They are supposed to illustrate an information need that often cannot be satisfied by traditional systems, but for which a new technique is proposed. Figure 6.1 lists some examples found in the literature (Ogle and Stonebraker, 1995; Moelaert El-Hadidiy et al., 1999; Westerveld, 2000; Picard, 1995). It is unclear whether these kinds of information needs are realistic. Below realistic needs are investigated by looking at what is depicted in images, and what types of images are searched for in realistic settings. The section concludes with a discussion on the topics used in evaluations.

What does an image show?

To come up with a realistic and diverse set of topics, a good starting point is to consider all types of information that can be conveyed by an image. Several studies in library science analysed this for deciding what to present in a manually constructed (textual) index. Most of these proposals are based on Panofsky's three levels of meaning, developed to describe art images (Panofsky, 1970). The three levels are *pre-iconography*, *iconography* and *iconology*. They correspond roughly to generic, specific and abstract meanings of an image. For example an image of the crucifixion of Christ will have pre-iconographic features (wooden cross, male figure), iconographic features (Jesus Christ, crucifixion) and iconologic features (religion, suffering). Markey (1988) adapted Panofsky's theory of meaning and built a scheme for indexing images (Table 6.1).

	Pre-Iconography (generic)	Iconography (specific)	Iconology (abstract)
Who?	kind of person kind of object	named person named object	mythical being
What?	kind of action kind of condition	named event	emotion abstraction
Where?	kind of geographical or architectural place	named geographic location	place symbolised
When?	cyclical time (season, time of day)	linear time (date, period)	emotion, abstraction symbolised by time

Table 6.1: Markey's subject analysis table

Shatford Layne (1986) distinguishes between *ofness*, the concrete and objective entities depicted (objects, places, actions), and *aboutness*, the abstract and subjective entities (feelings, symbolised concepts). Ofness corresponds to pre-iconographic and iconographic levels of meaning, aboutness to the iconologic level. Like Panofsky (1970) and Markey (1988), Shatford Layne (1986) acknowledges that an image can be at the same time specific and generic. For example, an image of the Empire State Building is at the same time an image of that specific building and an image of an arbitrary skyscraper.

What is searched for?

Although the studies presented above give a good overview of the kinds of information that is possibly present in images, for building good topics it is perhaps more useful to look at what people in realistic situations search for.

One of the largest investigations of visual information need is the HULTON study (Enser, 1993). In this study, over 2,500 requests to a picture archive (the HULTON DEUTSCH COLLECTION) were analysed and divided into two categories: unique requests of specific persons, events and locations ('Peter the Great', 'Trafalgar Square') and non-unique requests for generic concepts ('wanted-posters', 'dinosaurs'). This partition corresponds to the difference between Panofsky's iconographic and pre-iconographic levels of meaning. Enser found requests in both groups were often refined using restrictions on time, location or action ('1950s fridge', 'Edward VIII looking stupid'). He also found that most of the requests to the archive fell into the unique category (69%); about half of the requests were refined, mostly using time restrictions (34%).

Keister (1994) analysed requests to the Archive of the National Library of Medicine. She divided requests into visual requests, defined by what should be seen in the image, and topical requests, defined by background information. Keister found that the larger part of the requests was defined in terms of background and contextual information rather than in terms of the visual image content.

Markkula and Sormunen (2000) analysed illustration searching methods of journalists in one of the largest Finish newspapers. They found differences between searches for illustrations for different types of articles. To illustrate news articles, often current documentary photos were used, because there is little time to find the right image. Images for feature articles are more subjective and are needed less urgently. To illustrate these articles, often symbolic photos, or photos of themes are used. After generating illustration ideas, journalists could either search for photos themselves, or send a request to the archivist. Markkula and Sormunen found that the majority of the illustration needs (both searched for personally and requested from the archivists) are specific, i.e., most searches involved named objects or persons. About 20% of the requests involved types of objects ('animals', 'cow in the pasture', 'vehicles', 'a good photo of the front part of a bus'). Another important search type (33% of all search topics) was for documentary photos of recent news events. In this type of search, the contextual information is most important. Other, less frequent searches, involved named places and films and television programmes. Markkula and Sormunen report that for some topics journalists preferred personal searching and browsing over send-

VT0110: Find shots of a person diving into some water.

Figure 6.2: Example topic from the TRECVID2003 collection.

ing a request to an archivist. These were topics that were highly subjective ('young love'; 'a photo of a child's anxiety'), or thematic ('holidays in the south'; 'child care at home'). For information needs of these types, journalists found it was easier to look for photo's themselves, than to explain the topic to the archivist.

What is evaluated?

TRECVID adopts Enser (1993), and tries to balance requests for persons, things and events, as well as generic and specific requests. For example, Figure 6.2 shows a generic request for a shot of a person from the TRECVID2003 topic set. The TRECVID requests cover only the pre-iconography and iconography levels of Panofsky's classification, and ignore the iconology class. Arguably, the latter class is more difficult, since it involves abstract meanings. On the other side, the classes covered by TRECVID are representative, as the user studies show that these classes are the ones most commonly asked for. A comparison of the TRECVID2002 topics to requests submitted to British multimedia archives[4], shows that the same predominance over non-abstract types exists in both sets of requests (Smeaton and Over, 2003). A difference is that TRECVID has relatively many topics concerning generic persons or things, whereas the British archive requests cover more specific requests.

Evaluations based on COREL usually ignore these studies on image content and user behaviour. Instead, images from within the collection are used as queries. In a task-oriented view of evaluation, the themes in the COREL collection can be used as implicit task descriptions. For example, Figure 6.3 shows some example images (queries) from the *Thailand* theme. Each of these would ask for more images of Thailand. Often a single image per theme is used as a query(either the first image, or a random one). In the experiments in Chapter 4, we used all images from the collection as query

[4]BBC natural history unit, and BRITISH FILM INSTITUTE's national film and television archive.

Figure 6.3: Example queries from the COREL collection.

images. In a system-oriented setting, the implicit request is 'Find more images that are visually similar to this one'. For these types of requests, any image from the collection, or even from outside the collection, can be used. However, as the next section explains, it is convenient to use images from within the collection, since then cheap relevance judgements are available.

6.3.3 Relevance judgements

Once a set of documents and a set of topics are available, the only thing needed to complete the test collection are the relevance judgements. They indicate which documents are relevant to which topics. Sometimes, this is called the ground truth. We prefer the term relevance judgements, since that better reflects their nature: relevance judgements are the opinions of one or more users about what is relevant and what is not. Ground truth has a connotation of absoluteness, something not present in relevance judgements, since users often disagree on what is relevant and what is not. This disagreement, or subjectivity, is discussed at the end of this section. First, different types of relevance, and ways of obtaining relevance judgements are discussed.

Visual versus topical relevance

Since long, relevance has been an extensively studied concept, central in information retrieval. Mizzaro (1997) provides an annotated bibliography. Although *topicality*, or *aboutness* is the type of relevance focused on in most evaluations, many other criteria exist, like for example *recency*, *availability*, *credibility*, and *clarity*. (Borlund, 2003; Maclaughlin and Sonnenwald, 2002; Cosijn and Ingwersen, 2000; Schamber and Bateman, 1996; Barry and Schamber, 1998).

In image and video retrieval we can distinguish between two types of relevance in particular: topical relevance and visual similarity. Topical relevance means a document is on the same topic as the query, visual similarity means it looks the same as the query example(s). The latter concept is prob-

lematic, how does one decide if two things *look the same*? Santini (2001) distinguishes between *preattentive* similarity and *attentive* similarity. The former is the similarity that is experienced before the attention is focused, i.e., when we look at something as a whole, or at a glace. The latter form of similarity is at play after focusing attention and can be seen as an similarity with interpretation. When users search for visual material, they will most likely go beyond glancing at the retrieved result, thus attentive similarity is the type we will concentrate on. Theories of visual similarity exist (e.g., Tversky, 1977; Santini, 2001), but, in the end, it is the user who judges if two things are similar. Theories can help, but never replace the user.

Topical and visual similarity are useful in task-oriented and system-oriented tests respectively. Some content-based image retrieval papers argue for system-oriented testing and say visual similarity is what should be evaluated (Gunther and Beretta, 2000; Wenyin et al., 2001). In a task-oriented setting however, visual similarity is found to be of minor importance. Several user studies (Markkula and Sormunen, 2000; Choi and Rasmussen, 2002) have shown topicality to be the most important criterion for judging relevance of visual material. Only after topicality and contextual information ('how old is the image?', 'have I seen it before?'), were visual aspects considered. Thus, in a task-oriented test, topicality seems to be a more useful evaluation criterion than visual similarity. Nevertheless, it is important to keep in mind that we are dealing with information needs asking for visual data. If somebody is looking for images of James H. Chandler (VT0076), it is probably not only because they want some background information on this person; it is likely they want to retrieve images or video clips in which he is clearly visible. For background information, they could also search a textual collection. Thus, in a sense, the visual information need can be seen as placing constraints on the topically relevant shots. First of all, the shot has to be about James Chandler, and second, he has to be clearly visible. In fact, this second criterion carries two aspects: the visibility per se, and a quality judgement (the clarity). TRECVID judges only visibility per se, without taking quality into account. The assessor guidelines state (Over, 2004):

> When a topic says a shot must "contain x" that is short for "contain x to a degree sufficient for x to be recognisable as x to a human". This means among other things that unless explicitly stated, partial visibility or audibility may suffice.

Thus, the aboutness of a shot is not taken into account, nor is its quality. For a request for the Golden Gate bridge, a shot with a person discussing the bridge in detail is not considered relevant. The bridge must be visible, However, a shot with a small and hazy bridge in the background is marked

Figure 6.4: Keyframes from two shots marked relevant by TRECVID assessors for a Golden Gate bridge query (VT0083).

as relevant as a full pan of the bridge (see Figure 6.4). If somebody searches for visual material of the Golden Gate bridge, a hazy glimpse of the bridge in the background will not do, just like a brief mention of the bridge in a text document on San Francisco is not relevant for somebody looking for textual information on the Golden Gate bridge.

Obtaining relevance judgements

As indicated in Section 6.2, ideally full relevance judgements would be available, i.e., the relevance value for each document-topic pair would have been judged manually by some assessor. In practise, however, this is impossible. With a document collection of a realistic size it is unfeasible for somebody to assess each document for a given topic. Below, the process of obtaining relevance judgements for COREL and TRECVID are discussed.

Corel A cheap way to get judgements for a whole collection is offered by the COREL collection. Instead of manually judging the relevance for each query-document pair, the classification into themes is used as the basis for the relevance judgements. Thus, a collection image is relevant to a query image if and only if the two images are from the same theme.

As explained in the previous section, in a task-oriented evaluation with COREL the themes can be seen as implicit descriptions of the information need. This indicates that the classification into themes can be used as a basis for judging relevance. Thus, all images in the *Thailand* theme are regarded relevant to each of the query images from Figure 6.3. However, these theme based judgements are incomplete. Other themes may contain more images of Thailand, and the images from Figure 6.3 could also be considered relevant for the themes *Indigenous people*, *Beaches*, and *Elephants* respectively. However, since these images are not classified into these themes,

retrieving the Taiwanese beach with boats given a query from the beach theme, would be considered an error. Thus, avoiding doing full relevance judgements has its price. Still, as discussed in Section 6.2, incompleteness of relevance judgements is not necessarily problematic.

In system-oriented evaluation the task is to retrieve visually similar documents. Therefore, judgements need to consider this aspect. But it is unclear how to obtain judgements that are purely based on visual similarity. It is impossible to ignore all topical aspects, and to judge visual similarity only. The themes in the COREL collection may be a good alternative. The fact that many themes are shot in batch at the same location, and thus have a high within-theme similarity, can be exploited. Themes like *horses* (see Figure 4.9) have high within-theme similarity and can be useful for system-oriented evaluation. In the light of a system-oriented evaluation, the fact that images from the horses theme are found when querying with just an image of the grass background (cf. Section 4.3.3 and Figures 4.16 and 4.16), can be considered a good result. The retrieved horse images are visually similar to the grass query, since the background is the same. We have to take into account however, that selecting only themes with high within-theme similarity for system tests introduces potential bias. Müller et al. (2002) show results are highly sensitive to the specific subset of COREL themes used, i.e., scores are higher when the hardest topics are removed. They see this as a problem, since this allows researchers to artificially influence scores. However, absolute scores are meaningless, and scores cannot be compared across collections. Removing topics from the test collection changes the collection, and thus invalidates the direct comparison of scores. Only comparisons on the same collection are meaningful. In fact, the different topic sets used by Müller et al. do not result in different relative rankings of approaches. The four approaches tested (zero, one, two, or three iterations of feedback) lead to the same conclusion regardless of the topic set used (more iterations is better). Still, the set of themes used for system-oriented testing potentially introduces bias. For example, it makes a difference whether the selected themes are mainly coherent in colour or in texture. But, if the selected themes reflect the desired system behaviour, and the number of selected themes is large enough, a subset of themes can be valuable for identifying the approach that best captures the type of similarity present in the selected themes.

TRECVID In TRECVID relevance judgements are created using a pooling method (see Section 6.2). Documents not in the pool are assumed to be not relevant. This assumption may not be valid, but for text retrieval this hardly influences the comparative results of approaches (see Section 6.2.3).

For content-based image and/or video retrieval, such tests of pool quality have not been performed yet. This section investigates the pool quality of the TRECVID2003 collection. We look at the influence of pool depth and we test for a possible bias against unjudged runs.

In TRECVID2003, for each topic the pool has been created by taking the top N results from each of the submissions. The resulting set of documents is then manually inspected for relevance. This set of documents together with the relevance judgement for each of these (either relevant or unrelevant), is known as the *qrels*. The pool depth N for TRECVID2003 was either 50 or 100, depending on the number of relevant documents found in the top 50 (if many were found, the depth was increased to 100). To test the effect of pool depth on the the measurements, we looked at smaller pool sizes. We re-evaluated all submissions on qrels obtained from pool depths varying from 1 to 50 (these modified qrels can easily be obtained by assuming all documents that are not retrieved within any top N for a given topic are unrelevant for that topic). Figure 6.5 shows the MAP for all submissions based on the original qrels (circles), and for the different pool sizes (dots), the submissions are sorted by decreasing original MAP. The figure shows, that the scores based on qrels from the smaller pools follow the trend of the original scores. This means the ranking of systems is not influenced much by the pool depth. We measure the correlations between the original MAPs for the submissions, and the MAPs obtained from other pool depths using Kendall's τ, a measure of the correlation between two rankings (e.g., Conovar, 1980). Figure 6.6 shows the scores obtained from smaller pools are highly correlated to the scores obtained from the original full pools. Even a pool depth of $N = 3$, (i.e., only the first three documents of each submission get evaluated), shows a correlation factor $\tau > .90$.

To test if there exists a potential bias against runs that did not contribute to the pool, we follow a similar approach. For each submission, we compute the MAP based on the original pool, and the MAP based on a modified pool from which we removed documents that are uniquely contributed by the submission under study. A third MAP was computed based on a modified pool from which we removed the documents uniquely contributed by that submission's group. Figure 6.7 shows for each submission the original MAP scores, and the ones obtained after removing that submission or the submission's group from the pool. The results based on the modified pools follow the original results almost perfectly. The correlation between original MAPs and modified MAPs very high $\tau > .99$

Modifications of the TRECVID2003 pool by looking at smaller pool depths or removing submissions or even entire groups from the pool hardly influences the comparative results, i.e., the ranking of submissions. This in-

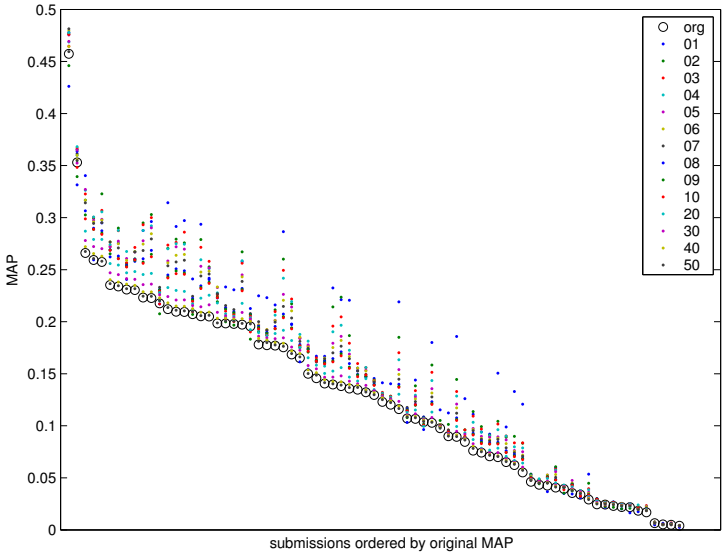


Figure 6.5: MAP for TRECVID2003 submissions based on original qrels and qrels obtained from smaller pool depths.

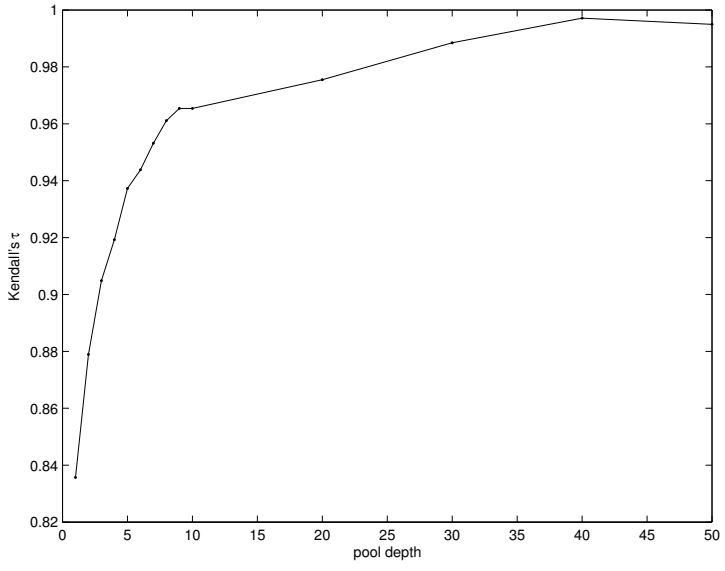


Figure 6.6: Kendall's τ between MAP obtained from different pool depths and MAP obtained from the original pool.

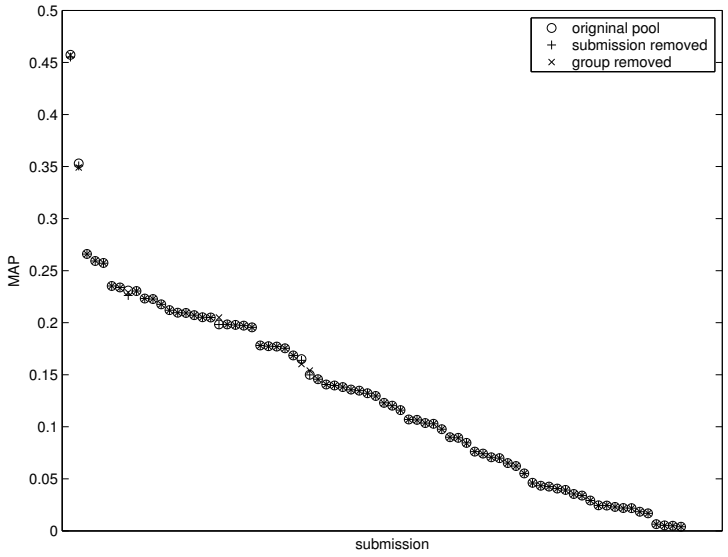


Figure 6.7: MAP for TRECVID2003 submissions based on original qrels and qrels after removing submission or submission's group from the pool.

dicates that the pool is of good quality, and thus a useful tool for further evaluation experiments.

Subjectivity of judgements

Section 6.2.3 discussed the inherent subjectivity of judgements in text retrieval. Clearly, judgements in visual information retrieval are subjective as well. However, judging visibility as is done in TRECVID, is arguably more objective than judging aboutness or topicality in TREC. The requested item is either visible or not, there is little room for discussion. Thus, agreement on visibility in TRECVID can be expected to be relatively high. Experiments with multiple assessors judging the same topics are needed to verify this. Because disagreement between assessors has been found not to influence comparative results in text retrieval (see Section 6.2.3), also relative ranking of approaches in visual retrieval is expected to be stable with regard to the subjectivity of judgements on visibility.

Judgements based on topical relevance are potentially more problematic. It can be hard to decide whether an image is for example about *Thailand* or not. It is unclear how retrieval results would be influenced if assessments on topical relevance would go beyond the classification into themes.

Visual relevance is also highly subjective. While visibility is clearly defined, Squire and Pun (1997) show the agreement between human assessors

on visual similarity is far from absolute. But, it remains unclear how this influences retrieval results.

6.4 Multimedia retrieval performance

Following the laboratory test methodology for text retrieval, each evaluated approach produces a ranked list of documents for each topic in the test collection. The quality of the ranked lists is measured based on the positions of the relevant documents in the list. Section 6.4.1 discusses metrics that can be used to assess the quality of the ranked list. Section 6.4.2 discusses alternative

6.4.1 Metrics

Over the years many different metrics have been proposed for measuring text retrieval performance. A few measures have become common practise, the most important ones being non-interpolated mean average precision and recall-precision graphs. These measures have been critically analysed, and are found to be stable and valuable measures (Buckley and Voorhees, 2000) (see also Section 6.2.2). Although multimedia documents differ from textual documents, there is no reason to believe these measures are less valuable in the context of multimedia retrieval. The differences between the document types mainly influence the way topics are constructed and how relevance is judged (see Sections 6.3.2 and 6.3.3). Apart from the different types of documents, multimedia retrieval is, like text retrieval, about returning relevant information to a user. Thus, standard text retrieval metrics can be used. Still, in the field of content-based image retrieval, some of the traditional text retrieval evaluation measures are criticised, and new measures are proposed. (Müller et al., 2001) gives an overview of measures proposed for multimedia retrieval. In this section, the metrics that are not common in text retrieval are reviewed.

Gunther and Beretta (2000) argue it is important to have a single performance metric to compare content-based information retrieval systems on. Additional measures can be presented, but a primary measure is needed to avoid comparison and claims based on arbitrary measures. However, they ignore the standard metric used in text retrieval, mean average precision, and propose a new one. The metric they propose is the normalised ranking, calculated as the mean of the ranks of the relevant documents retrieved within the top N, with a penalty for not retrieved documents. The measure is normalised to values between 0 and 1, and averaged over multiple queries.

The normalised ranking is weakly related to (mean) average precision, as it has precision and recall-oriented sides (the ranks at which documents are retrieved are taken into account, and a penalty is given for non-retrieved documents). The difference is that with average precision the influence of newly retrieved documents decreases as we go down the ranking, i.e., the first relevant documents found contribute most to the final score. With the normalised ranking metric, all documents contribute equally.

Huijsmans and Sebe (2003) criticise common text retrieval measures, and in particular the use of recall-precision graphs, because these graphs hide differences between topics. Huijsmans and Sebe claim that the variance in the number of relevant documents across topics has a large influence on retrieval results. They introduce the notion of *generality*, the fraction of relevant documents in the collection, and propose evaluation measures that include this notion. The proposed measures however, are set-based measures, where the set size depends on the generality of the topic. A disadvantage of set-based measures is that the ranks at which relevant documents are retrieved are ignored. Thus retrieving relevant documents at the top of the ranked list gives the same scores as retrieving them at the bottom of this set. Also, focusing on differences between topics of different generality, and thus considering single topic scores rather than averages, makes measures unreliable (Voorhees, 1998).

De Vries et al. (2004a) criticise the use of predefined retrieval units in video retrieval. They argue users are not necessarily interested in shots, but want *video clips*, i.e., pieces of video that may or may not coincide with shots. Dropping the assumption of predefined retrieval units introduces the problem of matching returned items to relevant items, since these matches may not be exact. To solve the problem De Vries et al. introduce the notion of *tolerance to irrelevance*, the amount of time the user is willing to spend on watching irrelevant material before reaching the relevant information. Based on this notion, they develop a measure similar to the expected search length (Cooper, 1968). It measures how much time the user has to spend viewing non-relevant material before the desired amount of relevant material is found. In the proposed alternative, systems are expected to return entry points in the video stream rather than shots or fragments. Assessors start watching from the returned entry points until relevant material is found, or until the tolerance to irrelevance time is exceeded. When relevant material is found, the assessors mark both beginning and end points. These relevant fragments are then used to compute the expected search length.

The first edition of TRECVID has also experimented with non-predefined retrieval units. However, there traditional recall and precision based measures have been used, based on the matches between relevant and retrieved

items. A relevant item is matched if the overlap between relevant and retrieved items is large enough. Since the second edition, TRECVID has used predefined common shots as the retrieval unit. This sidesteps the problems with overlap, and simplifies pooling. However, often relevant shots cluster in time, and an entire scene consisting of multiple shots may be relevant. Because of the predefined retrieval units systems can get rewarded multiple times for being able to find the same relevant information.

6.4.2 Informal analysis of results

Although measures are important for comparing approaches, they do not provide any insight into why one approach is better than another. Also measures often can not distinguish between approaches, i.e., the difference in scores may not be statistically significant. For example, the mean average precision scores on TRECVID2002 and TRECVID2003 of the content-based approaches (i.e., the ones that do not use speech transcripts, or other textual information) are all low (between .025 and .029, see Tables 4.10 and 4.11, and do not differ significantly. Thus, based on the mean average precision scores we can only conclude the approaches are equally good (or equally bad). This does not imply the approaches retrieve the same type of documents. It just means the approaches have comparable performance *on average*. To learn when one approach is better and when the other, one can look at individual topic scores, but still this gives little insight. To get a better feeling of what different approaches can and cannot do, it is useful to visually inspect retrieval results. At a glance this may seem like redoing relevance judgements, but we are not only interested in the retrieved relevant documents, but also in retrieved, non-relevant documents, and in not retrieved, relevant documents. In fact, a careful analysis of these mistakes may provide more insight than looking at the good results only.

The following aspects are worth inspecting.

- How is image similarity captured? I.e., how do the *top* retrieved documents correspond to the query (irrespective of being relevant or not)?
- How did the retrieved documents contribute to the scores? I.e., how do the *relevant* retrieved documents correspond to the query?
- Why are relevant documents missed? I.e., how do the not-retrieved relevant documents differ from the retrieved ones?

To investigate these aspects, the top retrieved documents need to be inspected as well as the relevant documents. Chapter 4 showed many exam-

ples of such inspections, see for example Figures 4.14 and 4.15. Such analyses have proved useful in explaining results.

6.5 Discussion

This chapter discussed the requirements for and problems in evaluating multimedia retrieval. We focused on controlled evaluation in laboratory tests in the CRANFIELD tradition, and distinguished two types of laboratory settings: task-oriented and system-oriented.

We found COREL to be mainly useful for a system-oriented evaluation because of its clear organisation in coherent themes. The fact that some of the themes appear to be shot in batch at the same location introduces a homogeneity within the theme that cannot be expected among relevant documents in a media archive setting. This makes COREL less appropriate for a task-oriented setting. In system-oriented testing homogeneous themes are useful, because they provide a ground truth for visual similarity. It is however unclear if this notion of visual similarity corresponds to human judged visual similarity. Furthermore, selecting themes that are suitable for testing visual similarity can easily introduce bias. COREL is a set of themes that may be suitable for system-oriented testing, but for fairness and comparability a commonly agreed upon subset of themes is needed.

The TRECVID collections are on their way to becoming widely used test collections for video retrieval. The collections aim at task-oriented evaluation, and are indeed growing toward realistic sizes. The limited domain of the collections, and especially the limited number of sources, introduces the risk of over-fitting. Therefore, it is unclear how results on TRECVID transfer to other collections. Having only produced, broadcast video in the collection introduces a bias for speech and text oriented approaches. Indeed results on TRECVID2002 and TRECVID2003 show text oriented approaches outperform content-based approaches by far: not only for the approach described in this thesis (see Section 4.4, Table 4.13), but also for other approaches (Smeaton and Over, 2003; Smeaton et al., 2003a). Bias is not a problem if text approaches are compared to text approaches, content-based approaches to content-based approaches, and mixed approaches to mixed approaches. Therefore it is important that users of the TRECVID test sets clearly indicate what kind of approach they used.

Judgements in TRECVID are binary, i.e., a shot is relevant if and only if the requested item is visible. It may be useful to distinguish between visible and clearly visible, since only the latter type is usable in a realistic (production) setting where the retrieved visual material is to be re-used. There-

fore, it would be useful to introduce graded relevance judgements (highly-relevant, relevant, irrelevant), as sometimes used in text retrieval (Järvelin and Kekäläinen, 2000; Voorhees, 2001).

In the previous chapters we have used COREL for system-oriented testing. Although the subset used was not selected to maximise the visual similarity within themes, the main aim of the COREL based experiments was to test whether the models give intuitive results, i.e., to test whether they do capture visual similarity. The TRECVID collections have been used to evaluate the usefulness of generative probabilistic models in a realistic setting. Both collections have been used to provide illustrating examples.

Visually inspecting retrieval results as introduced in Section 6.4.2 has proved valuable. Apart from giving insight in the effects of using different approaches, e.g., homogeneous queries find visually similar documents, it also revealed artifacts of the collections, good results on COREL are sometimes due to background similarity.

7

Conclusions

Three issues in ad hoc retrieval from heterogeneous multimedia collections have motivated the research in this thesis:

- How can generative probabilistic models be applied to multimedia retrieval?
- Can we identify and leverage parallels between the use of generative models for multimedia retrieval and similar approaches to text retrieval?
- How do the techniques based on generative models perform on the task of ad hoc retrieval from a generic collection?

We have studied these issues by developing variants of generative probabilistic models and conducted experiments to evaluate their performance on the task of ad hoc retrieval from a heterogeneous multimedia archive. The main findings regarding the three research questions are summarised in Sections 7.1, 7.2 and 7.3 respectively. Section 7.4 identifies interesting areas for future research.

7.1 Generative probabilistic models for multimedia retrieval

How generative models can be used for disclosing multimedia archives has been studied in Chapters 3 and 5. Chapter 3 described a basic approach to modelling content and showed how to use the models in a multimedia retrieval setting. Chapter 5 investigated extensions of the basic content models and variants of the basic retrieval mechanism.

The basic approach builds a generative model for each document in the collection and uses the likelihood of the query conditioned on each of the models for ranking. Thus, document models are ranked by decreasing probability of generating the query. The main assumption is that the models that are likely to generate the query are those estimated from relevant documents.

We have shown that the generative modelling approach can be applied to visual documents, textual documents and combinations of the two. Visual information can be modelled using Gaussian mixture models, whereas so-called language models (effectively multinomial distributions) are suitable for modelling textual data. When a query contains both modalities, the joint likelihood of generating the query text and the visual query example(s) can be used for ranking.

The basic approach to finding model parameters is to use maximum likelihood estimates. However, the resulting models do not explain unseen events (query terms that did not occur in the document). Moreover, we have shown that retrieval results may be distorted by the influence of common events (terms or feature vectors that occur in many documents). These problems can be solved by interpolating the maximum likelihood estimates with a more general distribution, the so-called background distribution (Section 3.6). This technique, known as smoothing, proved vital for good performance. An alternative to arrive at more accurate estimates is to take a Bayesian approach (see Section 5.4).

Chapter 5 has shown that the generative models are flexible tools for the representation and comparison of documents and queries. The chapter has discussed and used variants and extensions of the generative probabilistic models and has shown they all fit into a single probabilistic framework.

7.2 Parallels with language modelling

The second research issue, the identification and leverage of parallels with language modelling, has been addressed in many places throughout this thesis. This section highlights the parallels between our work in multimedia retrieval and previous generative modelling approaches to text retrieval.

First of all, the main ideas underlying the generative approaches in the two fields are the same: a query is treated as an observation from a relevant document model. This way, retrieval is treated as a classification problem; the goal is to find the generating source for a query. To implement this idea, both in text retrieval and in multimedia retrieval, probabilistic models are build to capture the characteristics of the documents. The likelihood of a query given each of these models is then used to rank the documents.

The technique of smoothing the maximum likelihood estimates is known to be crucial for text retrieval using generative models. For image retrieval, it turned out to be equally important to use a mixture of generic (background) and specific (foreground) models. The interpolated model reduces the influence of common events on retrieval results (Sections 3.6.2 and 4.2.3).

Many of the variants proposed in Chapter 5 have their counterparts in text retrieval. We have pointed out their relationship to probabilistic latent semantic indexing (pLSI) and latent Dirichlet allocation (LDA), models that have been developed as generative models for textual data, as well as the relationship between the models that use smoothing during training and the parsimonious language models that have been used in text retrieval.

Chapter 5 started with a probabilistic framework that covers the query generation and document generation variants of the generative models. This framework, in which the main question regards the probability of relevance given a query and a document, is at the basis of probabilistic approaches that have been proposed for text retrieval. Lafferty and Zhai (2003) have shown that the framework suits traditional probabilistic approaches as well as the modern language modelling techniques. In Section 5.7, we have shown that also the variants of generative models that we have proposed for multimedia retrieval fit in the framework.

7.3 Evaluation results

Experimentation has played an important role in the work presented here.

Thorough evaluation of the proposed models on the task of ad hoc retrieval from generic heterogeneous archives has shown that current content-based visual retrieval systems are not fit for this task. Scores for runs in which the available textual data (from ASR transcripts) has not been exploited tend to be much lower than scores for runs that use only text. This is the case for the techniques proposed in this thesis, but also for approaches tested by other research groups that participated in TRECVID. This does not mean content-based visual retrieval is useless; it only shows that on average text-based techniques give better results. However, some topics perform better when only visual data is used. Using both modalities, i.e., computing the joint likelihood of textual and visual query for ranking, combines the best of two worlds and gives better results than using either alone.

We have experimented with several ways of treating multiple examples in a single query, viz. let the user select a single representative example, compute the joint likelihood for all examples, combine results from separate examples in a round robin fashion and build a query model from the examples

and employ the document generation approach. The experiments have shown that round robin combination and document generation, strategies that treat multiple example queries as OR-queries (*Find documents that look like A* OR *B,* OR *C*) give the best results.

Using manually selected regions instead of full examples as queries has not shown to be useful in the query generation variant (Section 4.3.3). In document generation though, the combination of text results with topic model results has shown the best performance for topic models built from manually selected regions (Section 5.2).

Automatically selecting distinguishing samples to build the models from, as we have done in the variant that used background probabilities during training (Section 5.3) harmed the query generation results. However, the corresponding document generation variant outperformed all other query generation and document generation variants.

7.4 Directions for future research

Content-based multimedia retrieval has been studied for over a decade. The large-scale evaluations that have been carried out over the last few years give insight into the performance of current techniques on the task of ad hoc retrieval from heterogeneous multimedia collections. One lesson is that content-based techniques still perform poorly on this task. There is a lot of room for improvement. This section lists promising directions for future research.

7.4.1 Interactivity

Although this thesis has focused on approaches that require little user effort, we think that extending the scope to an interactive setting is necessary to improve query and document generation. We found that it is not possible for a user to produce the best query formulation in a single go, but with more interaction between user and system, it should be possible to obtain a better internal representation of the information need. Such interaction should go beyond the common iterations of relevance feedback in which a user indicates more relevant examples. In addition, there should be feedback in the other direction, from the system to the user. This feedback could explain the internal representation of the information need to the user, or the reason for retrieving documents. In the generative models setting, the system could for example present visualisations of the models or indicate

which parts of the query are well explained by the document models. More research is needed into this kind of interactivity between system and user.

7.4.2 Direct comparison of models

For ranking, query and document generation approaches use the likelihood of observations conditioned on models. Alternatively, one could compare models directly, using the cross-entropy, or KL-divergence between the model distributions. In fact, in the language modelling approach to text retrieval this seems to be the current trend (Zhai, 2002; Hiemstra et al., 2004; Lavrenko, 2004). Directly comparing models instead of computing the likelihood of observations has a number of advantages. It circumvents the choice between query generation and document generation variant, and it solves the problems that in the document generation variant the best matching document would be the document consisting of a repetition of a single data point. Therefore, it would be useful to directly compare models. However, the KL-divergence between two mixtures of Gaussians is not analytically solvable. Vasconcelos (2000) proposed an approximation, but it builds upon unrealistic assumptions (see Chapter 5) and does not perform well on retrieval from heterogeneous multimedia collections. More research is needed to find alternative approximations.

7.4.3 Evaluation methodology

Chapter 6 briefly reflected upon the use of common text retrieval evaluation methodology for evaluating retrieval approaches in a multimedia environment. A much broader study in this area is needed.

In designing evaluation methodology for multimedia retrieval, it is important to identify the users and tasks for which the techniques are developed. Ultimately, the techniques should be usable in practise. To test the usefulness of the techniques in a practical situation, operational tests are needed. In such tests, the techniques should aid users in performing a specific task. The effect of the techniques on the performance of the task can be seen as a measure of the usefulness of the techniques, but also factors like user satisfaction can be reported.

Current multimedia retrieval techniques are seldom mature enough to be tested in an operational setting, but also in system or task oriented tests, like the ones described in this book, it is important to keep this ultimate goal in mind. Therefore, it is important to identify areas in which (content-based) multimedia retrieval techniques can play a role. Test collections and measures for laboratory tests of multimedia retrieval systems have to be

carefully designed with real users and real application areas in mind. Only then results from the laboratory can be useful in practise.

A

Notation

Fonts

$\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$	boldface for vectors (lowercase) and matrices (uppercase)
$\mathcal{S}$, $\mathcal{D}$	calligraphic font for sets and collections
COREL,TRECVID	smallcaps for test collections (and other named entities)

Functions and relations

$\propto$	proportional to
$\equiv$	is defined as
$\mathrm{P}(\cdot)$	probability Mass function
$\mathrm{p}(\cdot)$	probability Density function
$\boldsymbol{A}^T$	transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	inverse of matrix $\boldsymbol{A}$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	multivariate Gaussian (or normal) distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathrm{mult}(\boldsymbol{\phi})$	multinomial distribution with parameter $\boldsymbol{\phi}$
RSV (d)	retrieval status value of document d

Symbols

$\mathcal{V} = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_S}\}$	visual document; a bag of samples
$\mathcal{T} = \{\mathrm{term}_1, \ldots, \mathrm{term}_N\}$	textual document; a bag of terms
$\boldsymbol{t} = (t_1, \ldots, t_T)$	alternative representation of textual document; a vector of term counts

$\boldsymbol{\theta} = (\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_{N_C}})$	parameters for a Gaussian mixture model with N_C components
$\boldsymbol{\phi} = (\phi_1, \ldots, \phi_{N_V})$	parameters for a language model (multinomial distribution) with N_V terms
$\boldsymbol{\phi}_{\mathrm{ML}}$	language model based on maximum likelihood estimates
$\boldsymbol{\phi}_{\mathrm{BG}}$	language model based on background probabilities
λ, λ_{Shot}, λ_{Scene}, λ_{Coll}	mixing parameters for language models
κ	mixing parameter for image models
τ	Kendall's correlation coefficient

Bibliography

Abberley, D., Renals, S., and Cook, G. (1998). Retrieval of broadcast news documents with the THISL system. In *Proc IEEE ICASSP*, pages 3781–3784, Seattle.

Allan, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft, W. B., Dumais, S., Fuhr, N., Harman, D. K., Harper, D. J., Hiemstra, D., Hofmann, T., Hovy, E., Kraaij, W., Lafferty, J., Lavrenko, V., Lewis, D., Liddy, L., Manmatha, R., McCallum, A. K., Ponte, J. M., Prager, J., Radev, D., Resnik, P., Robertson, S., Rosenfeld, R., Roukos, S., Sanderson, M., Schwartz, R. M., Singhal, A., Smeaton, A. F., Turtle, H., Voorhees, E. M., Weischedel, R., Xu, J., and Zhai, C. (2003). Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, University of Massachusetts Amherst, September 2002. *SIGIR Forum*, 37(1):31–47.

ANP Beeld (2004). website. `http://www.anp.nl/beeld`.

Baan, J., Ballegooij, A. van, Geusebroek, J.-M., Hiemstra, D., Hartog, J. den, List, J., Snoek, C. G. M., Patras, I., Raaijmakers, S., Todoran, L., Vendrig, J., Vries, A. P. de, Westerveld, T., and Worring, M. (2001). Lazy users and automatic video retrieval tools in (the) lowlands. In Voorhees and Harman (2002).

Bakker, E. M., Huang, T. S., Lew, M. S., Sebe, N., and Zhou, X. S., editors (2003). *Image and Video Retrieval, Second International Conference, CIVR 2003, Urbana-Champaign, IL, USA, July 24-25, 2003, Proceedings*, volume 2728 of *Lecture Notes in Computer Science*. Springer.

Barnard, K., Duygulu, P., Forsyth, D., Freitas, N. de, Blei, D. M., and

Jordan, M. I. (2003). Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135.

Barnard, K., Duygulu, P., Freitas, N. de, and Forsyth, D. (2002). Object recognition as machine translation - part 2: Exploiting image database clustering models. unpublished, see also part 1: Duygulu et al. (2002).

Barry, C. L. and Schamber, L. (1998). Users' criteria for relevance evaluation: a cross-stiuational comparison. *Information Processing & Management*, 34(2/3):219–236.

Beaulieu, M., Baeza-Yates, R., Myaeng, S. H., and Järvelin, K., editors (2002). *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland. ACM Press.

Belongie, S., Carson, C., Greenspan, H., and Malik, J. (1998). Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the Sixth International Conference on Computer Vision*.

Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In Callan et al. (2003).

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Borlund, P. (2003). The concept of relevance in IR. *Journal of the American Society for Information Science and Technology*, 54(10):913–925.

Bosch, H. G. P., Vries, A. P. de, Nes, N., and Kersten, M. L. (2001). A case for image quering through image spots. In *Storage and Retrieval for Media Databases 2001*, volume 4315 of *Proceedings of SPIE*, pages 20–30, San Jose, CA, USA.

Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers*. Dover Publications.

Brown, P. F., Cocke, J. C., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Buckley, C. and Voorhees, E. M. (2000). Evaluating evaluation measure stability. In Yannakoudakis et al. (2000), pages 33–40.

Buckley, C. and Walz, J. (2000). The TREC-8 query track. In *Proceedings of the Eighth Text Retrieval Conference, TREC-8*. NIST Special Publications.

Callan, J., Cormack, G., Clarke, C., Hawking, D., and Smeaton, A. F., editors (2003). *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto Canada. ACM Press.

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Croft et al. (1998), pages 335–336.

Carson, C., Belongie, S., Greenspan, H., and Malik, J. (2002). Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8).

Choi, Y. and Rasmussen, E. M. (2002). User's relevance criteria in image retrieval in american history. *Information Processing & Management*, 38(5):695–726.

Cleverdon, C. W. (1967). The cranfield tests on index language devices. *Aslib Proceedings*, pages 173–192.

Clough, P. and Sanderson, M. (2003). The CLEF 2003 cross language image retrieval task. In Peters, C., editor, *Working Notes for the CLEF 2003 Workshop*.

Conovar, W. J. (1980). *Practical Non-Parametric Statistics*, pages 249–250. John Wiley and Sons.

Cooper, W. S. (1968). Expected search length: A single measure of retrieval effectiveness based on the weak ordering of retrieval systems. *American Documentation*, 19(1):30–41.

Cooper, W. S. (1988). Getting beyond boole. *Information Processing & Management*, 24(3):243–248.

Cooper, W. S. (1991). Some inconsistencies and misnomers in probabilistic information retrieval. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 57–61. ACM Press.

Cornacchia, R., Ballegooij, A. van, and Vries, A. P. de (2004). Gaussian mixture models in MonetDB. In *First International Workshop on Computer Vision meets Databases*.

Cosijn, E. and Ingwersen, P. (2000). Dimensions of relevance. *Information Processing & Management*, 36(4):533–550.

Croft, W. B. (1993). Knowledge-based and statistical approaches to text retrieval. *IEEE Expert: Intelligent Systems and Their Applications*, 8(2):8–12.

Croft, W. B. (2003). Salton award lecture - information retrieval and computer science: an evolving relationship. In Callan et al. (2003), pages 2–3.

Croft, W. B., Moffat, A., Rijsbergen, C. J. van, Wilkinson, R., and Zobel, J., editors (1998). *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press.

Cutting, D., Kupiec, J., Pedersen, J., and Sibun, P. (1992). A practical part-of-speech tagger. In *Proceedings of Applied Natural Language Processing*, pages 133–140.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

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

Draper, S. W., Dunlop, M. D., Ruthven, I., and Rijsbergen, C. J. van, editors (1999). *Proceedings of Mira 99: Evaluating Interactive Information Retrieval*, Electronic Workshops in Computing, Glasgow, Scotland. British Computer Society.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience, 2 edition.

Dunlop, M. (2000). Reflections on mira: interactive evaluation in information retrieval. *Journal of the American Society for Information Science*, 51(14):1269–1274.

Duygulu, P., Barnard, K., Freitas, N. de, and Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Seventh European Conference on Computer Vision*, pages 97–112.

Enser, P. G. B. (1993). Query analysis in a visual information retrieval context. *Journal of Document and Text Management*, 1(1):25–52.

Everitt, B. S. and Hand, D. J. (1981). *Finite Mixture Distributions*. Chapman and Hall.

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. (1997). Query by image and video content: the QBIC system. In Maybury (1997), pages 7–22.

Fuhr, N. (1992). Probabilistic models in information retrieval. *Computer Journal*, 35(3):243–255.

Garofolo, J., Auzanne, G., and Voorhees, E. M. (2000). The TREC spoken document retrieval track: A success story. In *Content-Based Multimedia Information Access, RIAO 2000 Conference Proceedings*.

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

Gemert, J. van (2003). Retrieving images as text. Master's thesis, Intelligent sensory information systems, University of Amsterdam.

Girolami, M. and Kabán, A. (2003). On an equivalence between PLSI and LDA. In Callan et al. (2003), pages 433–434.

Goldberger, J., Gordon, S., and Greenspan, H. (2003). An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures. In *International Conference on Computer Vision (ICCV 2003)*.

Greenspan, H., Goldberger, J., and Mayer, A. (2002). A probabilistic framework for spatio-temporal video representation & indexing. In Heyden, A., Sparr, G., Nielsen, M., and Johansen, P., editors, *Computer Vision* -

ECCV 2002, 7th European Conference on Computer Vision, volume 2353 of *Lecture Notes in Computer Science*, pages 461–475. Springer.

Greenspan, H., Goldberger, J., and Mayer, A. (2004). Probabilistic space-time video modeling via piecewise GMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):384–396.

Greenspan, H., Goldberger, J., and Ridel, L. (2001). A continuous probabilistic framework for image matching. *Computer Vision and Image Understanding*, 84(3):384–406.

Gunther, N. J. and Beretta, G. (2000). A benchmark for image retrieval using distributed systems over the internet: BIRDS-I. Technical Report HPL-2000-162, HP Laboratories.

Harman, D. K. (1996). Overview of the fourth text retrieval conference (TREC-4). In *Proceedings of the Fourth Text Retrieval Conference, TREC-4*, pages 1–23. NIST Special Publications.

Harter, S. P. (1996). Variations in relevance assessments and the measurement of retrieval effectiveness. *Journal of the American Sociecty for Information Science*, 47(1):37–49.

Hauptmann, A., Baron, R. V., Chen, M.-Y., Christel, M., Duygulu, P., Huang, C., Jin, R., Lin, W.-H., Ng, T., Moraveji, N., Papernick, N., Snoek, C. G. M., Tzanetakis, G., Yang, J., Yang, R., and Wactlar, H. D. (2003). Informedia at TRECVID 2003: Analyzing and searching broadcast news video. In Smeaton et al. (2003b).

Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In Nicolaou, C. and Stephanidis, C., 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.

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

Hiemstra, D., Robertson, S., and Zaragoza, H. (2004). Parsimonious language models for information retrieval. In *Proceedings of the 27th Annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185. ACM Press.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM Press.

Hoiem, D., Sukhankar, R., Schneiderman, H., and Huston, L. (2003). Object-based image retrieval using the statistical structure of images. Technical Report IRP-TR-03-13, Intel.

Huijsmans, D. P. and Sebe, N. (2003). Extended performance graphs for cluster retrieval. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2001)*, pages 26–31.

Ianeva, T., Vries, A. P. de, and Westerveld, T. (2004). A dynamic probabilistic retrieval model. In *IEEE International Conference on Multimedia and Expo (ICME)*. to appear.

Järvelin, K. and Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In Yannakoudakis et al. (2000), pages 41–48.

Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. MIT Press.

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

Jeon, J., Lavrenko, V., and Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. In Callan et al. (2003).

Jin, X. and French, J. C. (2003). Improving image retrieval effectiveness via multiple queries. In *Proceedings of the first ACM international workshop on Multimedia databases*, pages 86–93. ACM Press.

Jong, F. M. G. de, Gauvain, J.-L., Hiemstra, D., and Netter, K. (2000). Language-based multimedia information retrieval. In *Content-Based Multimedia Information Access, RIAO 2000 Conference Proceedings*, pages 713–722.

Jordan, M. I. (2003). Graphical models. *Statistical Science*, 19(1):140–155. Special Issue on Bayesian Statistics.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing - An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall.

Kalt, T. (1998). A new probabilistic model of text classification and retrieval. Technical Report UM-CS-1998-018, University of Massachusetts, Computer Science Department.

Keister, L.-H. (1994). User types and queries: Impact on image access systems. In Fidel, R., Hahn, H. B., Rasmussen, E. M., and Smith, P. J., editors, *Challenges in Indexing Electronic Text and Images*, pages 7–22.

Kraaij, W. (2004). *Variations on language modeling for information retrieval*. PhD thesis, University of Twente.

Kraaij, W., Westerveld, T., and Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In Beaulieu et al. (2002), pages 27–34.

Kraft, D. H., Croft, W. B., Harper, D. J., and Zobel, J., editors (2001). *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press.

Lafferty, J. and Zhai, C. (2003). Probabilistic IR models based on document and query generation. In Croft, W. B. and Lafferty, J., editors, *Language Modeling for Information Retrieval*, volume 13 of *The Kluwer International Series on Information Retrieval*. Kluwer Academic Publishers.

Lavrenko, V. (2004). *A generative theory of relevance*. PhD thesis, Graduate school of the Univeristy of Massachusetts Amherst.

Lavrenko, V. and Croft, W. B. (2003). Relevance-based language models: Estimation and analysis. In Croft, W. B. and Lafferty, J., editors, *Language Modeling for Information Retrieval*, volume 13 of *The Kluwer International Series on Information Retrieval*. Kluwer Academic Publishers.

Lavrenko, V., Manmatha, R., and Jeon, J. (2004). A model for learning the semantics of pictures. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA. MIT Press.

Leung, C. H. C. and Ip, H. H.-S. (2000). Benchmarking for content-based visual information search. In Laurini, R., editor, *Advances in Visual Information Systems, 4th International Conference, VISUAL 2000, Lyon,*

France, November 2-4, 2000, Proceedings, volume 1929 of *Lecture Notes in Computer Science*. Springer.

Lew, M. S., Sebe, N., and Eakins, J. P., editors (2002). *Image and Video Retrieval, International Conference, CIVR 2002, London, UK, July 18-19, 2002, Proceedings*, volume 2383 of *Lecture Notes in Computer Science*. Springer.

Li, J. and Wang, J. Z. (2003). Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10).

Luo, H., Fan, J., Xiao, J., and Zhu, X. (2003). Semantic principal video shot classification via mixture gaussian. In *IEEE International Conference on Multimedia and Expo (ICME)*.

Maclaughlin, K. L. and Sonnenwald, D. H. (2002). User perspectives on relevance criteria: a comparison among relevant, partially relevant , and not-relevant judgements. *Journal of the American Society for Information Science and Technology*, 53(5):327–342.

Markey, K. (1988). Access to iconographical research collections. *Library Trends*, 37(2):154–174.

Markkula, M. and Sormunen, E. (2000). End-user searching challenges indexing practices in the digital newspaper photo archive. *Information Retrieval*, 1(4):259–285.

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

Marques, O. and Furht, B. (2002). *Content-based image and video retrieval*. Kluwer Academic Publishers.

Maybury, M. T. (1997). *Intelligent multimedia information retrieval*. MIT Press.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley-Interscience.

Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, 44:335–341.

Miller, D. R. H., Leek, T., and Schwartz, R. M. (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 214–221.

Mizzaro, S. (1997). Relevance: the whole history. *Journal of the American Society for Information Science*, 48(9):810–832.

Moelaert El-Hadidiy, F., Poot, H. J. G., and Velthausz, D. D. (1999). Multimedia information retrieval framework. In *Semantic Issues in Multimedia Systems*, volume 8 of *IFIP 2.6 Working Conference on Database Semantics*.

Müller, H. (2002). *User interaction and evaluation in content-based visual information retrieval*. PhD thesis, Computer Vision and Multimedia Laboratory, University of Geneva, Geneva, Switzerland.

Müller, H., Marchand-Maillet, S., and Pun, T. (2002). The truth about Corel – evaluation in image retrieval. In Lew et al. (2002).

Müller, H., Müller, W., Squire, D. M., Marchand-Maillet, S., and Pun, T. (2001). Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters (Special Issue on Image and Video Indexing)*, 22(5):593–601. H. Bunke and X. Jiang Eds.

Narasimhalu, A. D., Kankanhalli, M. S., and Wu, J. (1997). Benchmarking multimedia databases. *Multimedia Tools and Applications*, 4(3):333–356.

Natsev, A. and Smith, J. R. (2003). Active selection for multi-example querying by content. In *IEEE International Conference on Multimedia and Expo (ICME)*.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia object image library (coil-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University.

Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134.

NPS Klokhuis (2004). Omroep archief. TV broadcast. May 7, 2004.

Oard, D. W. and Dorr, B. J. (1996). A survey of multilingual text retrieval. Technical Report UMIACS-TR-96-19, Univ. of Maryland Institute for Advanced Computer Studies Report No. UMIACS-TR-96-19.

Ogilvie, P. and Callan, J. (2003). Using language models for flat text queries in XML retrieval. In *Proceedings of the Initiative for the Evaluation of XML Retrieval Workshop (INEX 2003)*.

Ogle, V. E. and Stonebraker, M. (1995). Chabot: retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48.

Ordelman, R. J. F. (2003). *Dutch Speech Recognition in Multimedia Information Retrieval*. Phd thesis, University of Twente, Enschede.

Over, P. (2004). personal communication.

Over, P. and Taban, R. (2002). The trec-2001 video track framework. In Voorhees, E. M. and Harman, D. K., editors, *The Tenth Text Retrieval Conference, TREC-2001*. NIST Special Publications.

Panofsky, E. (1970). *Meaning in the Visual Arts*. Penguin, London.

Pentland, A., Picard, R. W., and Sclaroff, S. (1996). Photobook: content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3):233–254.

Peters, C., Braschler, M., Gonzalo, J., and Kluck, M., editors (2003). *Advances in Cross-Language Information Retrieval: ThirdWorkshop of the Cross-Language Evaluation Forum, CLEF 2002*, volume 2785 of *Lecture Notes in Computer Science*.

Peters, C., Braschler, M., and Julio Gonzalo, M. K., editors (2002). *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001*, volume 2406. Springer-Verlag.

Picard, R. W. (1995). Toward a visual thesaurus. Technical Report TR 358, M. I. T. Media Laboratory.

Picard, R. W., Kabir, T., and Liu, F. (1993). Real-time recognition with the entire Brodatz texture database. In *Proceedings of the IEEE International Conference on Computer Vision*.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In Croft et al. (1998), pages 275–281.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Reid, N. H. (1999). The photographic collections in St. Andrews university library. *Scottish Archives*, 5:83–90.

Robertson, S. (1977). The probability ranking principle in IR. *Journal of Documentation*, 33:294–304.

Robertson, S. and Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall.

Rubinstein, Y. D. and Hastie, T. (1997). Discriminative vs informative learning. In *Knowledge Discovery and Data Mining*, pages 49–53.

Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual Communication and Image Representation*, 10(1):39–62.

Salton, G. (1968). *Automatic information organization and retrieval*. McGraw-Hill.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Santini, S. (2001). *Exploratory image databases, content-based retrieval*, chapter 4, pages 105–164. Academic Press.

Saracevic, T. (1995). Evaluation of evaluation in information retrieval. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 138–146. ACM Press.

Schamber, L. and Bateman, J. (1996). User criteria in relevance evaluation: Toward development of a measurement scale. In *Proceedings of the American Society for Information Science*, pages 218–225.

Schmid, C. (2004). Weakly supervised learning of visual models and its application to content-based retrieval. *International Journal of Computer Vision*, 56(1-2):7–16.

Sclaroff, S., La Cascia, M., and Sethi, S. (1999). Unifying textual and visual cues for content-based image retrieval on the world wide web. *Computer Vision and Image Understanding*, 75(1-2):86–98.

Shatford Layne, S. (1986). Analyzing the subject of a picture: a theoretical approach. *Cataloguing and Classification*, 6:39–62.

Sivia, D. S. (1996). *Data analysis, a Bayesian tutorial*. Oxford university press.

Smeaton, A. F., Kraaij, W., and Over, P. (2003a). TRECVID 2003 - an introduction. In Smeaton et al. (2003b).

Smeaton, A. F., Kraaij, W., and Over, P., editors (2003b). *TRECVID 2003 Workshop*, Gaithersburg, MD, USA. NIST, NIST Special Publications.

Smeaton, A. F. and Over, P. (2003). The TREC-2002 video track report. In Voorhees and Buckland (2003).

Smeaton, A. F., Over, P., Costello, C. J., Vries, A. P. de, Doermann, D., Hauptmann, A., Rorvig, M. E., Smith, J. R., and Wu, L. (2002a). The TREC-2001 video track: Information retrieval on digital video information. In *Research and Advanced Technology for Digital Libraries, 6th European Conference, ECDL 2002, Rome, Italy, September 16-18, 2002, Proceedings*, volume 2458 of *Lecture Notes in Computer Science*, pages 266–275, Rome, Italy. Springer.

Smeaton, A. F., Over, P., and Taban, R. (2002b). The TREC-2001 video track report. In Voorhees and Harman (2002).

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380.

Smith, J. R. (1998). Image retrieval evaluation. In *IEEE Workshop on Content-based Access of Image and Video Libraries*.

Smith, J. R. and Chang, S.-F. (1997). Querying by color regions using visualseek content-based visual query system. In Maybury (1997), pages 23–41.

Song, F. and Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 316–321. ACM Press.

Sormunen, E., Markkula, M., and Järvelin, K. (1999). The perceived similarity of photos-seeking a solid basis for the evaluation of content-based retrieval algorithms. In Draper et al. (1999).

Sparck Jones, K. (1974). Automatic indexing. *Journal of Documentation*, 30:393–432.

Sparck Jones, K. and Rijsbergen, C. J. van (1975). Report on the need for and provision of an "ideal" information retrieval test collection. Technical Report British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge.

Sparck Jones, K. and Rijsbergen, C. J. van (1976). Information retrieval test collections. *Journal of Documentation*, 32(1):59–75.

Sparck Jones, K., Robertson, S., Hiemstra, D., and Zaragoza, H. (2003). Language modeling and relevance. In Croft, W. B. and Lafferty, J., editors, *Language Modeling for Information Retrieval*, volume 13 of *The Kluwer International Series on Information Retrieval*. Kluwer Academic Publishers.

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

Sparck Jones, K. and Willett, P. (1997a). Evaluation. In Sparck Jones and Willett (1997c), chapter 4, pages 167–174.

Sparck Jones, K. and Willett, P. (1997b). Models. In Sparck Jones and Willett (1997c), chapter 5, pages 257–263.

Sparck Jones, K. and Willett, P., editors (1997c). *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc.

Squire, D. M., Müller, W., Müller, H., and Raki, J. (1999). Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In *The 11th Scandinavian Conference on Image Analysis*, pages 143–149, Kangerlussuaq, Greenland.

Squire, D. M. and Pun, T. (1997). A comparison of human and machine assessments of image similarity for the organization of image databases. In Frydrych, M., Parkkinen, J., and Visa, A., editors, *The 10th Scandinavian Conference on Image Analysis*, pages 51–58, Lappeenranta, Finland.

Tague-Sutcliffe, J. (1992). The pragmatics of information retrieval experimentation, revisited. *Information Processing & Management*, 28(4):467–490.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. Wiley Series in Probability and Mathematical Statistics. John Wiley and Sons.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4):327–352.

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons.

Vasconcelos, N. (2000). *Bayesian Models for Visual Information Retrieval*. PhD thesis, Massachusetts Institut of Technology.

Vasconcelos, N. and Lippman, A. (1998). 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.

Vasconcelos, N. and Lippman, A. (2000). A probabilistic architecture for content-based image retrieval. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR2000)*, pages 216–221.

Voorhees, E. M. (1998). Variations in relevance judgments and the measurement of retrieval effectiveness. In Croft et al. (1998), pages 315–323.

Voorhees, E. M. (2001). Evaluation by highly relevant documents. In Kraft et al. (2001), pages 74–82.

Voorhees, E. M. (2002). The philosophy of information retrieval evaluation. In Peters, C., Braschler, M., and Julio Gonzalo, M. K., editors, *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001*, volume 2406, pages 355–370. Springer-Verlag.

Voorhees, E. M. and Buckland, L. P., editors (2003). *The Eleventh Text Retrieval Conference, TREC-2002*. National Institute of Standards and Technology, NIST Special Publications.

Voorhees, E. M. and Buckland, L. P., editors (2004). *The Twelfth Text Retrieval Conference, TREC-2003*. National Institute of Standards and Technology, NIST Special Publications.

Voorhees, E. M. and Harman, D. K. (2000). Overview of the eighth text retrieval conference (TREC-8). In *Proceedings of the Eighth Text Retrieval Conference, TREC-8*. NIST Special Publications.

Voorhees, E. M. and Harman, D. K., editors (2002). *The Tenth Text Retrieval Conference, TREC-2001*, volume 10. National Institute of Standards and Technology, NIST Special Publications.

Vries, A. P. de, Kazai, G., and Lalmas, M. (2004a). Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *RIAO 2004 Conference Proceedings*, pages 463–473.

Vries, A. P. de, Westerveld, T., and Ianeva, T. (2004b). Combining multiple representations on the TRECVID search task. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*.

Wallace, G. K. (1991). The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44.

Wenyin, L., Su, Z., Li, S., Sun, Y.-F., and Zhang, H. (2001). A performance evaluation protocol for content-based image retrieval algorithms/systems. In *IEEE CVPR workshop on Empirical Evaluation Methods in Computer Vision*.

Westerveld, T. (2000). Image retrieval: Content versus context. In *Content-Based Multimedia Information Access, RIAO 2000 Conference Proceedings*, pages 276–284.

Westerveld, T. (2002). Probabilistic multimedia retrieval. In Beaulieu et al. (2002), pages 437–438.

Westerveld, T., Hiemstra, D., and Jong, F. M. G. de (2000). Extracting bimodal representations for language-based image retrieval. In *Multimedia '99, Proceedings of the Eurographics Workshop*, pages 33—42. Springer.

Westerveld, T., Ianeva, T., Boldareva, L., Vries, A. P. de, and Hiemstra, D. (2003a). Combining information sources for video retrieval. In Smeaton et al. (2003b).

Westerveld, T. and Vries, A. P. de (2003a). Experimental evaluation of a generative probabilistic image retrieval model on 'easy' data. In *Proceedings of the Multimedia Information Retrieval Workshop 2003*. in conjunction with the 26th annual ACM SIGIR conference on Information Retrieval.

Westerveld, T. and Vries, A. P. de (2003b). Experimental result analysis for a generative probabilistic image retrieval model. In Callan et al. (2003).

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

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

Witbrock, M. J. and Hauptmann, A. (1998). Speech recognition for a digital video library. *Journal of the American Society for Information Science*, 49(7):619–632.

Yan, R., Hauptmann, A., and Jin, R. (2003). Multimedia search with pseude-relevance feedback. In Bakker et al. (2003), pages 238–247.

Yannakoudakis, E., Belkin, N. J., Leong, M.-K., and Ingwersen, P., editors (2000). *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press.

Zaragoza, H., Hiemstra, D., and Tipping, M. (2003). Bayesian extension to the language model for ad hoc information retrieval. In Callan et al. (2003), pages 4–9.

Zhai, C. (2002). *Risk Minimization and Language Modeling in Text Retrieval*. PhD thesis, School of Computer Science, Carnegie Mellon University.

Zhai, C. and Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In Kraft et al. (2001), pages 334–342.

Zhu, L., Rao, A., and Zhang, A. (2002). Theory of keyblock-based image retrieval. *ACM Trans. Inf. Syst.*, 20(2):224–257.

Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments? In Croft et al. (1998), pages 307–314.

Author index

Subject index

Summary

This thesis discusses information retrieval from multimedia archives. Multimedia archives are collections of documents containing a mixture of text, images, video and audio. This work focuses on documents containing visual material and investigates search and retrieval in collections of images and video, where video is defined as a sequence of still images. No assumptions are made with respect to the content of the documents: the collections are not restricted to a specific domain (e.g., images of fingerprints or collections of x-ray pictures). Instead we concentrate on retrieval from generic, heterogeneous multimedia collections. In this research area a user's query typically consists of one or more example images and the implicit request is: "Find images similar to this one." In addition the query may contain a textual description of the information need. The research presented here addresses three issues within this area.

First, we show how generative probabilistic models can be applied to multimedia retrieval. For each document in the collection a probabilistic model is built: a statistical description of the document's characteristics. For each of these models we then compute the probability that the query is generated from the model and we show the documents corresponding to the models with the highest probability to the user. The assumption is that these are the most relevant documents, i.e., those with characteristics corresponding to the query characteristics. Visual information is modelled using Gaussian mixture models and information derived from language (e.g., the speech of the video soundtrack) is modelled using statistical language models. This dissertation presents different ways of using the generative probabilistic models for multimedia retrieval and shows that they all fit in a common probabilistic framework. In addition we show that it is important to distinguish between common characteristics, shared by most documents, and distinguishing characteristics, those specific to one document. This distinction can be made

either by reducing the influence of common characteristics in the query, a technique known as smoothing, or by building probabilistic models that put more emphasis on the distinguishing elements in a document.

The second issue addressed is the parallel between the use of generative probabilistic models for multimedia retrieval and comparable models for text. In the area of text retrieval, generative models have been studied intensively in the last couple of years. This thesis describes how the techniques developed for language relate to the multimedia techniques presented here and how these parallels can be leveraged. An example of a language modelling technique that is known to be essential in text retrieval, is the aforementioned smoothing. This thesis shows that the same technique is crucial to image retrieval.

Third, this thesis studies evaluation. A large part of this thesis is dedicated to experimentation. We tested the model variants using a number of collections including the test collections of TRECVID, the international workshop series for benchmarking video retrieval. On average, language-based approaches outperform approaches based on visual information. However, for some queries visual information is important. A combination of both modalities gives the best results when searching a heterogeneous multimedia collection.

Samenvatting

Dit proefschrift gaat over het zoeken van informatie in multimedia collecties: verzamelingen van documenten die behalve tekst ook andere media bevatten, zoals beeld, geluid en video. Dit werk concentreert zich op documenten met visuele informatie en behandelt het doorzoeken van collecties met stilstaande beelden en video, waarbij video wordt gezien als een reeks opeenvolgende stilstaande beelden. Op voorhand worden geen aannames gemaakt over de inhoud van de collecties. Er wordt dus niet gezocht in een collectie met een specifieke inhoud, zoals bijvoorbeeld een verzameling foto's van vingerafdrukken of een collectie röntgenfoto's. We kijken naar het doorzoeken van algemene, heterogene beeldcollecties. Een zoekvraag van een gebruiker bestaat uit één of meer voorbeeldplaatjes en de impliciete opdracht is: "Vind meer plaatjes zoals deze.". Eventueel kan de zoekvraag worden aangevuld met een tekstuele beschrijving van de informatiebehoefte. Dit proefschrift behandelt drie onderwerpen binnen dit probleemgebied.

Ten eerste wordt bekeken hoe generatieve kansmodellen kunnen worden toegepast voor het zoeken in multimedia collecties. Dat kan door voor elk document in de collectie een kansmodel te bouwen: een statistische beschrijving van de kenmerken van het document. Voor elk van de modellen wordt vervolgens berekend wat de kans is dat de zoekvraag gegenereerd wordt uit dat model. De documenten die corresponderen met de meest waarschijnlijke modellen worden aan de gebruiker getoond. De aanname is dat dit de meest relevante documenten zijn, dat wil zeggen de documenten waarvan de kenmerken het best overeenkomen met de zoekvraag. Visuele informatie wordt gemodelleerd met Gaussian mixture modellen; voor op taal gebaseerde informatie (bv. de spraak behorend bij een video) worden statistische taalmodellen gebruikt. Dit proefschrift presenteert verschillende varianten van het gebruik van generatieve kansmodellen voor het zoeken in multimedia collecties en laat zien dat alle varianten binnen hetzelfde probabilistische

raamwerk vallen. Verder laten we zien dat het van belang is onderscheid te maken tussen algemene kenmerken, die door de meeste documenten gedeeld worden, en onderscheidende kenmerken, die specifiek zijn voor een bepaald document. Dat onderscheid kan op verschillende manieren gemaakt worden: door minder gewicht toe te kennen aan de algemene elementen in de zoekvraag, een techniek die smoothing genoemd wordt, of door meer nadruk te leggen op de onderscheidende kenmerken bij het bouwen van een statistische beschrijving van een document.

Het tweede onderwerp in dit proefschrift is de relatie tussen de generatieve kansmodellen voor multimedia documenten en vergelijkbare modellen voor tekst. Over het gebruik van taalmodellen voor het zoeken in tekstuele collecties is al veel bekend. Dit proefschrift brengt in kaart hoe de voor tekst ontwikkelde technieken zich verhouden tot de hier gepresenteerde technieken voor multimedia en hoe deze dwarsverbanden benut kunnen worden. Een voorbeeld van een taalmodeltechniek waarvan het belang voor zoeken in tekst al langer bekend is is het hierboven genoemde smoothing. Deze techniek blijkt nu ook cruciaal voor het doorzoeken van beeldmateriaal.

Het derde onderwerp is evaluatie. Een niet onbelangrijk deel van dit werk bestaat uit het experimenteren met verschillende varianten van de ontwikkelde modellen. De deelname aan TRECVID speelt hierin een belangrijke rol, omdat deze internationale serie workshops voor de evaluatie van multimedia zoeksystemen de mogelijkheid biedt technieken te testen in een raamwerk dat gemodelleerd is naar een realistische setting. In het algemeen doen op beeld gebaseerde technieken onder voor technieken die gebruik maken van tekstuele informatie. Toch betekent dit niet dat beeldtechnieken geen zinnige bijdrage kunnen leveren. Ook al levert zoeken op basis van tekst gemiddeld genomen betere resultaten, er zijn zoekvragen waarvoor beeld uitblinkt. Het benutten van beide modaliteiten geeft de beste resultaten bij het zoeken van informatie in een multimedia collectie.

Curriculum Vitae

Thijs Westerveld was born on 23 September 1974 in Gendringen, the Netherlands. In 1992, he graduated from secondary school (VWO) at "Christelijke Scholengemeenschap Aalten". He then started his studies in computer science at the University of Twente in Enschede and received a M.Sc. degree in 1997 after completing a thesis on the correction of spelling errors in dialogue systems. From 1997 to 2002, he worked as a research assistant in the language, knowledge and interaction group at the university of Twente, where he actively participated in a number of national and European projects in the areas of text retrieval and multimedia retrieval. In 2002, he accepted a position at the centre for mathematics and computer science (CWI) in Amsterdam where he worked on the Waterland project. Within this project, he continued his work on the use of probabilistic models for multimedia retrieval.

Thijs has published in international journals and conferences on information retrieval and multimedia retrieval and recently received the best paper award in the conference on image and video retrieval (CIVR2004). He has served as a program committee member for several international information retrieval conferences and he has been a reviewer for international journals in the areas of information retrieval and pattern recognition. Since September 2004, he has worked as a scientific researcher at the centre for mathematics and computer science in Amsterdam.

SIKS dissertation series

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games

1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets

1999-3 Don Beal (UM)
The Nature of Minimax Search

1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems

1999-7 David Spelt (UT)
Verification support for object database design

1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coup (EUR)
Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations, Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

2001-1 Silja Renooij (UU) Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management: The Role of Mental Models in Business Systems Design

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology; Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM) Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses

2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM)
Learning Search Decisions

2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes

2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politile gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining

2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM)
Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (CWI)
Using generative probabilistic models for multimedia retrieval