

University of London
Imperial College of Science, Technology and Medicine
Department of Computing

**Statistical Models for
Semantic-Multimedia Information Retrieval**

João Miguel Costa Magalhães

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, September 2008

Abstract

This thesis addresses the problem of improving multimedia information retrieval by exploring semantic-multimedia analysis. For this goal we researched two complementary search paradigms: (1) search-by-keyword and (2) search-by-semantic-example.

Search-by-keyword produces excellent results and users are completely familiarised with this type of search on the Web. The user is already “educated” to express his/her ideas with a sequence of keywords that summarize the sought information. In the search-by-keyword paradigm one needs to be able to detect the presence of concepts (keywords) in multimedia content. In our approach, for each possible query keyword we estimate a statistical model based on multimedia features that were pre-processed. More concretely, we studied the of family linear regression models to estimate the model of each keyword in a multi-modal feature space. The unique continuous multi-modal feature-space is created using a minimum description length criterion to find an optimal feature-space representation.

Unfortunately not all concepts or ideas can be described by keywords: a user might have a “creative idea” for which he/she can only supply some examples. It is in these situations that search-by-semantic-example comes to the rescue of the user. With this search paradigm the user formulates a query with a “semantic example” that hints at the semantics that he/she wants to find. Then, the semantic multimedia information retrieval system searches the multimedia database by evaluating the semantic similarity between the query and the previously indexed multimedia. This semantic comparison of two multimedia documents is the central problem of search-by-semantic-example. Thus, we investigated the two main aspects of this problem: similarity metrics and ways to reduce the semantic space complexity.

Our achievements can be divided into quantitative and qualitative aspects. On the quantitative side, experiments with different collections showed that the proposed statistical framework can deliver an excellent trade-off between precision, scalability, and flexibility. On the qualitative side, we were able to contribute to a better understanding of how to take advantage of semantics in multimedia retrieval systems by processing it at the two extremes of the information chain: at the content side and at the user side.

Acknowledgments

I would like to start by thanking the people that made this thesis possible: my supervisor Stefan R uger for his support, encouragement and wise words; the *Funda  o para a Ci ncia e Tecnologia* that funded me during the first three years of my PhD under the scholarship SFRH /BD /16283 /2004 /K5N0; Nicu Sebe and Joemon Jose for the challenging and stimulating viva.

My colleagues Peter, Alexei, Simon, Paul, Daniel and Ed, in the Multimedia and Information Systems research group at Imperial, were excellent research peers in the lab and excellent friends outside the lab, e.g., at PAFC games, at the Greyhound races, or at Opal. My colleagues in room 433 “supported” me during the anti-stress moments at 4pm in the Level 8 bar: Uri, Alok, Georgia, Mohammad and Dave. Thanks to all of them!

This journey started in 2004 when I moved from the beautiful and sunny Lisbon to the exciting, cosmopolitan and gray London. I received much support from Portugal throughout this journey that now reaches a conclusion. I would like to send warm thanks to Anabela, Jorge, Vasco and my brother.

In London, an excellent way of killing the *saudades* was the small Portuguese community that would always provide me with a constant supply of *pasteis de nata*, *morcelas*, *ovos moles*, *bacalhan*, etc. Of course their support did not end in this multitude of healthy items and the companionship and friendship were the base stones of our small community. Thus, I would like to thank Catarina, Hugo, Silvestre, Antigoni, Jo o, Luisa, Ana, Rita, Catarina Almeida and Vanda for all their positive support.

Lisbon, Catarina and Lucilene are now my future – thank you for making me dream.

Finally, and most importantly, my last acknowledgement goes to my parents. I dedicate this thesis to them.

Contents

1	Introduction	13
1.1	Multimedia Information	15
1.2	User Information Needs	16
1.3	Multimedia Information Retrieval Systems	17
1.3.1	Multimedia Analysis	18
1.3.2	Indexing	19
1.3.3	Query Processing	20
1.3.4	Retrieval	21
1.4	Scope	21
1.4.1	High-Level Multimedia Analysis	22
1.4.2	Search-by-Keyword	23
1.4.3	Search-by-Semantic-Example	23
1.5	Contributions	23
1.6	Publications	24
1.7	Organization	26
2	Evaluation Methodologies	27
2.1	Introduction	27
2.2	Effectiveness	27
2.2.1	Defining Relevance	28
2.2.2	Precision and Recall	30
2.2.3	Metrics Generalization and Normalization	33
2.3	Efficiency	33
2.3.1	Indexing Complexity	33
2.3.2	Query Analysis Complexity	34
2.4	Collections	34
2.4.1	Text Collections	35
2.4.2	Image Collections	35
2.4.3	Video Collections	37
2.5	Collection Generation	38

2.5.1	Data Sampling Strategies	39
2.5.2	Data Complexity	39
2.5.3	Information Relevance	39
2.5.4	Generalization and Cross-Datasets	40
2.6	Summary	40
Part 1 Indexing Semantic-Multimedia		41
3	Semantic-Multimedia Analysis	42
3.1	Introduction	42
3.2	Single-Media Analysis	43
3.2.1	Image Analysis	43
3.2.2	Text Analysis	53
3.2.3	Audio Analysis	54
3.3	Multi-Modal Analysis	55
3.3.1	Structure Analysis	55
3.3.2	Heuristic based Models	57
3.3.3	Statistics based Models	58
3.4	Discussion	61
3.5	Conclusions	62
4	A Multi-Modal Feature Space	64
4.1	Introduction	64
4.2	Multi-Modal Keyword Models	66
4.3	Optimal Data Representation	68
4.3.1	Assessing the Data Representation Error	68
4.3.2	The MDL Principle	70
4.4	Dense Spaces Transformations	71
4.4.1	Visual Features Pre-Processing	72
4.4.2	Visual Transformation: Hierarchical EM	72
4.4.3	Experiments	75
4.4.4	Results and Discussion	76
4.5	Sparse Spaces Transformations	79
4.5.1	Text Feature Pre-Processing	79
4.5.2	Text Codebook by Feature Selection	80
4.5.3	Experiments	81
4.5.4	Results and Discussion	81
4.6	Conclusions and Future Work	82
4.6.1	Future Work	83

5	Keyword Models	84
5.1	Introduction	84
5.2	Keyword Baseline Models	85
5.2.1	Rocchio Classifier	85
5.2.2	Naïve Bayes Model	86
5.3	Keywords as Logistic Regression Models	88
5.3.1	Regularization	89
5.3.2	Maximum Likelihood Estimation	90
5.3.3	Large-Scale Model Computation	91
5.4	Relationship to other Approaches	92
5.4.1	Kernel Methods	92
5.4.2	pLSA	92
5.4.3	Generalized Linear Models	93
5.5	Evaluation	93
5.5.1	Collections	93
5.5.2	Experiment Design	94
5.5.3	Text-Only Models	95
5.5.4	Image-Only Models	98
5.5.5	Multi-Modal Models	101
5.6	Conclusions and Future Work	108
5.6.1	Retrieval Effectiveness	108
5.6.2	Model Selection	108
5.6.3	Computational Scalability	109
5.6.4	Semantic Scalability	109
5.6.5	Future Work	110
	Part 2 Searching Semantic-Multimedia	111
6	Searching Multimedia	112
6.1	Introduction	112
6.2	Content based Queries	114
6.3	Relevance Feedback	115
6.4	Semantic based Queries	116
6.4.1	Keyword based Queries	117
6.4.2	Natural Language based Queries	117
6.4.3	Semantic Example based Queries	117
6.4.4	Semantic Similarity	118
6.5	Summary	118

7	Keyword Spaces	120
7.1	Keywords and Categories	120
7.2	Defining a Keyword Space	122
7.3	Keyword Vectors Computation	125
7.3.1	Automatic Keyword Annotations	126
7.3.2	User Keyword Annotations	127
7.3.3	Upper and Lower Bounds	127
7.4	Querying the Keyword Space	128
7.5	Keyword Vectors Dissimilarity	128
7.5.1	Geometric Spaces	129
7.5.2	Histograms	131
7.5.3	Probabilistic Spaces	132
7.6	Evaluation	133
7.6.1	Collections	133
7.6.2	Experiments Design	134
7.6.3	Results and Discussion	135
7.7	Conclusions and Future Work	150
7.7.1	Future Work	151
8	Conclusion	152
8.1	Achievements	152
8.2	Future Work	154
8.3	Limitations of Semantic-Multimedia IR	155
8.4	New Challenges in Semantic-Multimedia IR	156
8.5	Influence on Multimedia Applications	157
	Nomenclature	159
	References	164

Figures

Figure 1.1. Information pipeline in IR applications.	14
Figure 1.2. A Web page can be seen as a multimedia document.	15
Figure 1.3. A video can be seen as a multimedia document.	16
Figure 1.4. A classic multimedia IR architecture.	17
Figure 1.5. Relation between information symbols and semantic abstraction.	18
Figure 1.6. The semantic-multimedia analysis process.	23
Figure 2.1. Retrieval effectiveness metrics based on relevant documents.	31
Figure 2.2. Interpretation of precision-recall curves.	32
Figure 2.3. Example of an ImageCLEF document.	37
Figure 3.1. Scope of the semantic-multimedia analysis problem.	42
Figure 3.2. Inference of single class models.	44
Figure 3.3. Translation models.	46
Figure 3.4. Two different types of random fields.	50
Figure 3.5. Knowledge based models.	53
Figure 3.6. Spatial structure of an HTML document, (Cai et al., 2003).	56
Figure 3.7. Temporal structure of a video document.	57
Figure 4.1. The traditional statistical learning theory framework.	66
Figure 4.2. Bias-variance trade-off curve.	69
Figure 4.3. Hierarchical EM algorithm.	75
Figure 4.4. Model selection for the Gabor filters features (Corel5000).	77
Figure 4.5. Model selection for the Tamura features (Corel5000).	77
Figure 4.6. Model selection for the marginal moments of HSV colour histogram features (Corel5000).	78
Figure 4.7. Model selection for the bag-of-word features (Reuters).	82
Figure 5.1. Form of the binomial logistic model.	89
Figure 5.2. Reuters-21578 retrieval MAP evaluation.	96
Figure 5.3. Reuters-21578 retrieval MP@20 evaluation.	96

Figure 5.4. Interpolated precision-recall curve evaluation on the Reuters-21578.	97
Figure 5.5. Retrieval precision for different space dimensions (text-only models).	97
Figure 5.6. Corel retrieval MAP for different keyword models.	98
Figure 5.7. Corel retrieval MP@20 for different keyword models.	99
Figure 5.8. Interpolated precision-recall curves for different keyword models.	99
Figure 5.9. Retrieval precision for different space dimensions.	101
Figure 5.10. MAP by different modalities (TRECVID).	102
Figure 5.11. MP@20 by different modalities (TRECVID).	102
Figure 5.12. Interpolated precision-recall curve for the text-only models (TRECVID).	104
Figure 5.13: Interpolated precision-recall curve for image-only models (TRECVID).	104
Figure 5.14. Interpolated precision-recall curve for multi-modal models (TRECVID).	105
Figure 5.15. Interpolated precision-recall curves for different modalities (LogisticRegL2).	105
Figure 5.16. Retrieval precision for different space dimensions (TRECVID, text-only).	106
Figure 5.17. Retrieval precision for different space dimensions (TRECVID, image-only).	107
Figure 5.18. Retrieval precision for different space dimensions (TRECVID, multi-modal).	107
Figure 6.1. Examples of search spaces of visual information.	113
Figure 6.2. The scope of semantic query-processing.	113
Figure 6.3. Semantic based search.	116
Figure 7.1. Commutative diagram of the computation of semantic similarity between two multimedia documents.	120
Figure 7.2. Example of Flickr images annotated with the keyword London.	122
Figure 7.3. A keyword space with some example images.	123
Figure 7.4. A multimedia document description.	124
Figure 7.5. Unit spheres for standard Minkowski distances.	130
Figure 7.6. MAP of the different dissimilarity functions (Corel Images).	136
Figure 7.7. MP@20 of the different dissimilarity functions (Corel Images).	136
Figure 7.8. Interpolated precision-recall curves of the different dissimilarity functions (Corel).	137
Figure 7.9. Interpolated precision-recall curves of the different dissimilarity functions (Corel).	137
Figure 7.10. MAP of the different dissimilarity functions (TRECVID).	138
Figure 7.11. MP@20 of the different dissimilarity functions (TRECVID).	138
Figure 7.12. Interpolated precision-recall curves of the different dissimilarity functions (TRECVID).	139

Figure 7.13. Interpolated precision-recall curves of the different dissimilarity functions (TRECVID).	139
Figure 7.14. Interpolated precision-recall curves of the different dissimilarity functions (Corel).	141
Figure 7.15. Interpolated precision-recall curves of the different dissimilarity functions (TRECVID).	141
Figure 7.16. Effect of user keywords accuracy on the MAP (Corel).	142
Figure 7.17. Effect user keywords accuracy on the MP@20 (Corel).	143
Figure 7.18. Effect of user keywords accuracy on the MAP (TRECVID).	143
Figure 7.19. Effect of user keywords accuracy on the MP@20 (TRECVID).	144
Figure 7.20. Effect of the number of concepts on the MAP (Corel).	146
Figure 7.21. Effect of the number of concepts on the MP@ 20 (Corel).	146
Figure 7.22. Effect of the number of concepts on the MAP (TRECVID)	147
Figure 7.23. Effect of the number of concepts on the MP@20 (TRECVID)	147
Figure 7.24. Example of image keyword-categories relationships.	149

Tables

Table 2.1. Summary of evaluation collections used in this thesis.	35
Table 5.1. MAP comparison with other algorithms (Corel).	100
Table 5.2. MAP comparison with other algorithms (TRECVID).	103
Table 7.1. Summary of collections used on the experiments.	133
Table 7.2. MAP for user keywords.	140
Table 7.3. Comparison between automatic keywords and user keywords.	145
Table 7.4. Semantic analysis performance per image.	148

Acronyms

ASR	Automatic Speech Recognition
DCT	Discrete Cosine Transform
DL	Description Length
EM	Expectation-Maximization
GLM	Generalized Linear Models
GMM	Gaussian Mixture Model
IG	Information Gain
IR	Information Retrieval
JS	Jensen-Shannon
KL	Kullback-Leibler
LDA	Latent Dirichlet Allocation
LLSF	Linear Least Squares Fit
LSA	Latent Semantic Analysis
MAP	Mean Average Precision
MDL	Minimum Description Length
MI	Mutual Information
MIR	Multimedia Information Retrieval
MP@20	Mean Precision at 20
pLSA	Probabilistic Latent Semantic Analysis
SVD	Singular Value Decomposition
SVM	Support Vector Machine

1

Introduction

“Humans get data about events using the five senses – vision, sound, touch, taste and smell. We assimilate this data with previous knowledge, both external and internal, to experience an event. ... Humans first developed languages and then invented different mechanisms to propagate knowledge derived from their experience.... We’ve developed different mechanisms (ranging from the written language, print, photographs, telegraph, telephone, radio, television and now Internet) for people to share experience across time.”

*Ramesh Jain, “Knowledge and Experience,”
IEEE Multimedia 2001.*

Human knowledge is by far the richest multimedia storage system. Language and other communication mechanisms, e.g., facial expressions, can only express a small part of one’s experiences and knowledge (Jain 2001). Vision and hearing, the most used senses during communication, carry a great part of the experience or knowledge that we wish to share. Information captured by these two human senses can also be effectively and efficiently captured, stored and processed by computers – everyone has collections of his/her holiday pictures, karaoke songs, videos etc. For these recorded experiences to be shared, some mechanism must be able to interpret human queries, and retrieve the closest match. For example, if users search their collection using a keyword or a phrase such as “door” or “door bell” they will expect the computer to return all relevant items. However, in most cases their search results in disappointment. This might be rooted in two reasons: (1) the context in which users formulate their information need is too vague and require users to refine their information need; and (2) the weak and blurred link between information representation schemes and the human semantic queries. These missing links are called the semantic gap (Smeulders et al. 2000).

Mechanisms that fill the semantic gap have yet to be fully understood. Computer algorithms that

extract low-level measures from visual streams (e.g., histograms, shape, motion) and sound streams (e.g., volume, pitch) are widely researched, providing a wide set of features that can be used to index multimedia. These multimedia low-level measures rely on data-driven features, which may be unrelated to the concepts expressed in the semantic query. The extraction of semantic information from multimedia content is a research topic that tries to mimic the way human perception works, and therefore is highly related to artificial intelligence. However, human perception is still not being understood at a level that we can imitate in a computational system.

An Information Retrieval (IR) system storing and delivering multimedia information is affected by this research problem when matching the semantics of the query with the semantics of the multimedia information (see Figure 1.1). On one hand the system must mimic human perception and extract the relevant semantics from the stored information and on the other hand the system must be able to interpret the human request and match it to the relevant stored information. The main problem when we wish to search our multimedia collections by expressing some semantic query is the missing relation between low-level features and human knowledge, or the *semantic gap*.

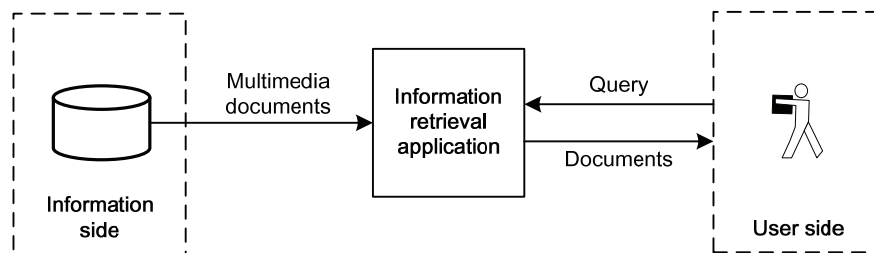


Figure 1.1. Information pipeline in IR applications.

Nowadays, applications that make use of semantics on the **information side** depend on manual annotations and other information extracted from the surrounding content, e.g., HTML links in case of Web content. This way of extracting multimedia semantics is flawed and costly. Doing the entire process automatically or even semi-automatically, can greatly decrease the operational and maintenance costs of such applications. In the information pipeline depicted in Figure 1.1 the **user query** is interpreted by the IR application with the same system that can simulate human perception to process the query and match it to the most relevant information. Thus, the semantic gap problem exists on both extremes of this pipeline.

It is in this scope that we propose a semantic-multimedia information extraction framework that offers a certain degree of semantic information processing capabilities. Thus, the main goal of this thesis is to **enhance multimedia retrieval applications by investigating new paradigms for searching semantic-multimedia information**. It is in this scope that we look at the global problem of semantic multimedia information retrieval and address its three main elements: semantic multimedia analysis, semantic query analysis and semantic matching. This contrasts with

previous work that has put the emphasis on only one or two of the mentioned aspects. With this approach we achieve a better understanding of the problem and identify the bottlenecks and the strengths of semantic multimedia information retrieval.

1.1 Multimedia Information

On the information side in Figure 1.1 multimedia documents contain a large amount of information that an IR system has to process and manage. Document formats can vary widely according to the usage domain, for example some communities consider a multimedia document to be a Web page, others a Flash presentation or a video file. These broadly different understandings of what a multimedia document is force us to define the notion of **syntax** and **semantics** of multimedia documents. One can easily identify the common characteristic across all listed examples as the presence of at least two different basic data types: text, image, video, speech, audio, synthetic data, interactive elements (links, events on user action, mouse over, open new window etc.) and structural elements (text formatting, images and video location etc.).

The image shows a screenshot of the Wikipedia article titled "Multimedia". The page layout includes a top navigation bar with options like "article", "discussion", "edit this page", and "history". A prominent banner at the top right states "22,332 have donated" with a "Donate now!" button. The main content area features a "Contents (hide)" table of contents with sections such as "1 Categorization", "2 Features", "3 Terminology", "4 Usage", and "5 Structuring information in a multimedia form". Below the table of contents, there is a section titled "Multimedia is a combination of content forms:" which displays six icons representing different media types: Text (document icon), Audio (headphones), Still Images (camera), Animation (film reel), Video (video camera), and Interactivity (hand cursor icon). The page also includes a sidebar with navigation links, a search box, and a toolbox.

Figure 1.2. A Web page can be seen as a multimedia document.

The syntax of a multimedia document is an aggregation of several elements from different data types that provide rich information and enhanced experience: the visible and audible data types are text, images, graphics, video and audio; structural elements are not visible by themselves; they determine the spatial and temporal organization of the other data types; interactive elements provide a way for the user to interact with the content.

Looking at the Web page example of Figure 1.2 and at the video example of Figure 1.3, one can see that humans segment documents into manageable blocks of information to later form a complete understanding of the document: we employ a sequential divide and conquer technique. Thus, in this thesis we define the syntax of multimedia documents as blocks of text-image pairs carrying some semantic information. As an example of semantic information one can examine the first segment of Figure 1.3 and identify it as a surf scene, as a well as having strong blue tones. It is exactly this semantic information that we want to capture and make accessible to applications. Thus, for the purposes of this thesis, we define the semantics of multimedia as a set of symbols (tokens) related to human understanding, (e.g., “surf scene”), and senses (e.g., blue tones).

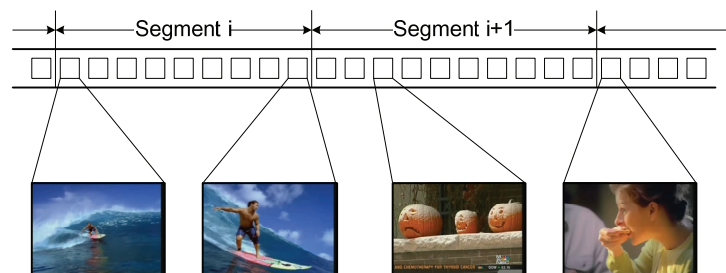


Figure 1.3. A video can be seen as a multimedia document.

Note that a text-image pair can be made of different combinations of the same image with different segments of text, and vice-versa. This simple definition of the syntax of multimedia documents allows us to cover both video and Web pages documents. The segmentation of the documents into pairs of (text; image) is left outside this thesis. Naphade et al. (1998) provide a good example of a video temporal segmentation algorithm and Yu et al. (2003) provide a good example of a Web page visual segmentation algorithm.

1.2 User Information Needs

The user side in Figure 1.1 depicts a generic paradigm of Information Retrieval: the user submits some information need and the system supplies the (hopefully) required information. Unlike text documents, multimedia documents do not necessarily contain symbols that the user can use to express his/her information need. This problem has roots in two different aspects. The first one is the richness of the searched information: visual information can communicate a wide variety of messages and emotions; audio content can also communicate feelings and emotions; structure also gives a different organization and usability (or user experience) to communication. In other words, multimedia documents give more freedom to the semantic interpretation of the communicated message.

The second aspect is the communication gap between the user and the system: computational systems can only process mathematical and logic expressions, and not all humans have the same skills at expressing ideas, emotions and feelings with those expressions.

Several techniques were developed and researched to empower the user with new tools to express his/her query that achieve a better mapping between what the user can express, what the system can extract from multimedia, and what the system can successfully match. I will present these different retrieval paradigms in the following section.

1.3 Multimedia Information Retrieval Systems

Information processing and management systems have existed for several decades. Most of the systems deployed until the mid 90s supported text data based documents while other types of data were largely left outside this forest of information retrieval systems. From all this experience a set of elementary functional modules were common to most systems: (1) an analysis module that extracts a vocabulary¹ from documents; (2) an indexing module to make documents efficiently accessible through its information symbols; (3) a query processing module to translate the user information needs into information symbols; and (4) the retrieval module to rank the stored documents according to the similarity between information symbols.

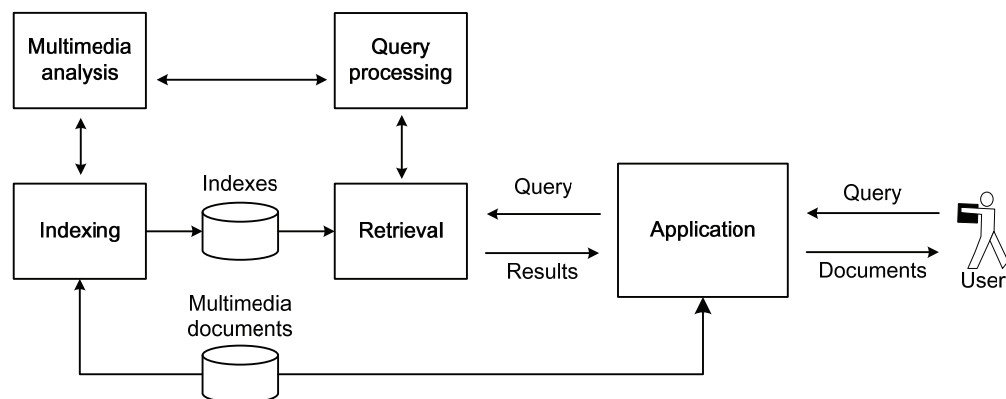


Figure 1.4. A classic multimedia IR architecture.

A multimedia information retrieval system, as the one depicted in Figure 1.4, is functionally similar to traditional IR systems but it has a small difference that impacts all algorithms present in other modules: the multimedia analysis algorithms produce information tokens that are not compatible with the ones produced by text analysis algorithms. Despite this fact, the architecture depicted in Figure 1.4 is still a good reference of a generic information retrieval system. We will

¹ This is also commonly known as *index tokens* in the context of indexing, and *feature vectors* in the context of document analysis.

now detail the modules of this generic architecture.

1.3.1 Multimedia Analysis

IR systems must analyse multimedia documents and extract features measuring the importance of information symbols. The objective of extracting these features is twofold:

- to associate multimedia documents to meaningful symbols of information that a human can search for
- to quickly locate relevant documents through an index of information symbols

These information symbols can be obtained automatically, semi-automatically or manually. While an automatic method executes an analysis task without the intervention of a human, semi-automatic methods includes a human as part of the analysis task. Note that some information can only be added by a human, as for example the name of a person, or the relation between two persons, e.g., friends. Different strategies are more adequate to the particular information domain that is being considered. For example, both Flickr² and Google's³ page rank rely on human edited information to improve search results: Flickr allow the user to tag images with some keywords that can be used for later searching those images; Google's algorithm rely on human edited links that point to the Web page being analysed to adjust its importance. One can say that Flickr's approach is semi-automatic and Google's approach is automatic because it relies on previously existing information.

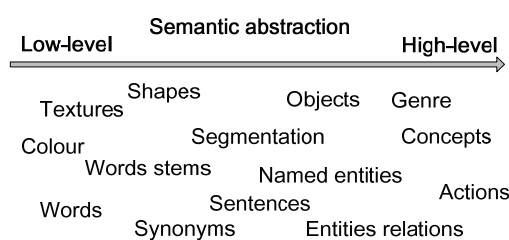


Figure 1.5. Relation between information symbols and semantic abstraction.

Figure 1.5 illustrates how we position some features in an imaginary scale of semantic abstraction: it ranges from low-level features to high-level features. Low-level features, such as a histogram of words or a colour histogram, are easily extracted by automatic methods. However, high-level features, such as topic of a news article or concepts represented in an image require more complex analysis algorithms due to the semantic dimension they involve.

² <http://www.flickr.com/>

³ <http://www.google.com/>

Traditional text analysis algorithms produce a limited set of features, e.g., occurring words, which contrast with multimedia analysis algorithms that produce a score, e.g., a probability, for all possible features (information symbols). This creates a dense high-dimensional vector of all features for all existing documents, which causes several problems such as storage space. Another (critical) difference is that the score associated to the extracted feature or information symbol has an error associated to it. This shows how the output of multimedia analysis algorithms will impact the entire multimedia IR system forcing us to use different techniques and algorithms to address the same problems.

Low-Level Multimedia Analysis

Low-level multimedia analysis directly extracts features from multimedia documents that are related to human senses or language, e.g., images colour, images texture, audio rhythm and words. These features are well studied and most of them have been developed in the area of data compression that exploits the characteristics of the human vision and hearing senses, e.g., JPEG and MP3. These low-level features are the information symbols that the system uses to build the index of multimedia documents.

High-Level Multimedia Analysis

High-level multimedia analysis aims to extract information that can be inferred from a multimedia document even if that information is not explicitly detectable by a computer. This involves some sort of prior-knowledge about the problem domain semantics, which is formally described with a set of concepts identified by keywords. These keywords capture part of the domain knowledge that can be used to infer the presence of a concept in a given multimedia document. These high-level features (keywords) are the index tokens that the system uses to build the index of semantic-multimedia documents.

1.3.2 Indexing

The information symbols extracted from multimedia content by the multimedia analysis algorithms are stored and managed by the indexing module. While the multimedia analysis algorithms impact the effectiveness of an IR system, the sole goal of the indexing module is to address the efficiency of the system. The core element of an indexing mechanism is the inverted-file index that lists information symbols and all documents containing that symbol.

Systems indexing multimedia information must employ a high-dimensional index to accommodate the high-dimensional data nature of multimedia information (as mentioned previously, multimedia analysis produce a score for all possible feature types and dimension). The efficiency of high-dimensional indexes is affected by several design aspects: compression of the

index reducing memory usage; tree-structured indexes or hash-based indexes allowing a quicker look-up of the index table; sorting documents of an index entry limits the number of analysed documents. An excellent reference discussing the efficiency of indexes is provided by (Baeza-Yates and Ribeiro-Neto 1999).

1.3.3 Query Processing

When a user submits a query the system must analyse the user input to transform it into the internal representation used to index multimedia information. Essentially, the query processing module must parse the user query according to a given query language, extract the information symbols contained in it, and pass it to the retrieval module to search the index for the matching documents. Most query languages support text queries while multimedia queries can be expressed with a variety of methods as we will describe next.

Sketch Retrieval

One of the first studied methods to query a multimedia database is sketch retrieval⁴. With this paradigm the user query is a visual sketch of what the user wishes to find; the system then processes this drawing to extract its features and searches the index for images that are visually similar. In this case the query processing has to extract the visual features that were used to index the visual information.

Search by Example

The previous method is somewhat limited because the algorithms cannot extract exactly the same type of features from both the visual sketch and the stored information. Thus, researchers came up with the possibility of allowing the user to submit an example image representing the information the user is searching for (Flickner et al. 1995). In this case the query processing has to extract the low-level features that were used to index the multimedia information. The wide range of different interpretations of an example, makes this approach more useful when the user provides more than one example to disambiguate the information need (Heesch 2005; Ortega et al. 1997).

Search by Keyword

Search-by-keyword is by far the most popular method of search query: the user describes his/her information needs with a set of keywords and the system just searches for the multimedia documents, see (Magalhães and Rügger 2007b) and (Yavlinsky 2007). One limitation of all high-level query/search methods is that the user can only submit keywords from a predefined vocabulary.

⁴The search engine RetrievR (<http://labs.systemone.at/retrievr/>) is an example of this approach.

Search by Semantic Example

Similarly to the search-by-example method, the user can also provide an example as a query but now it will be processed at the semantic level, e.g., returning a video with the same action or event (goal, football game). With this method the query processing has to extract the high-level features that were used to index the multimedia information (Magalhães, Overell and Rüger 2007). The problem in this case is the different interpretations that an example can have: the user can be looking for a particular object, e.g., a lion, a category of documents, e.g., safari, or an action, e.g., a lion eating a person.

Personalized/Adaptive Retrieval

The personalized/adaptive retrieval is a refinement to all other search methods – it explores the fact that the user has a search history and profile (Urban, Jose and Rijsbergen 2003; Magalhães and Pereira 2004). This extra information can improve the search experience by limiting information to particular domains or by limiting certain document formats or even transforming the multimedia documents into computationally less demanding versions.

1.3.4 Retrieval

The retrieval module is in charge of ranking documents according to their similarity to the user query. This module must navigate the index according to the information symbols contained in the input query to search for the most similar documents. A key aspect is the similarity metric that depends on the search space (e.g., colour, rhythm, words) and it ought to reflect human perception of similarity, see (Jose, Furner and Harper 1998; Heesch 2005; Howarth 2007; Yu et al. 2008).

1.4 Scope

In the previous section we presented a generic multimedia IR system as an overview of the research area in which this thesis is positioned. Previously, the definition of multimedia syntax and semantics, and the discussion on the user information needs have set the working domain of the present thesis. Within this scenario I have identified the main objective, namely to

enhance multimedia retrieval applications by investigating new paradigms for searching semantic-multimedia information.

We are now capable of isolating the relevant modules of Figure 1.4 and the core research problems that need to be addressed to accomplish this objective. Hence, the reach of the current research is limited to algorithms that can:

- **Improve user-query expressiveness:** algorithms must process both multimedia information and user query at a semantic level, thus increasing the level of information abstraction that multimedia IR systems can process;
- **Support different modalities:** algorithms must support different multimedia information, more specifically, they must process arbitrary text-image pairs as defined previously;
- **Low computational cost:** algorithms must be executed in a limited amount of time involving no noticeable delays to the user, and they must offer a good degree of computational scalability;
- **Good retrieval accuracy:** retrieved documents must be meaningful to the user query offering an improved user experience.

This list of requirements has obvious impacts on the **multimedia analysis** and **query processing** modules of the generic IR system architecture. All other modules depicted in Figure 1.4 are outside the scope of this thesis. Moreover, the required semantic expressiveness leads us into high-level analysis algorithms of semantic-multimedia information and into search paradigms where the user can express a query as a high-level abstraction of an information need (search-by-keyword and search-by-semantic-example).

1.4.1 High-Level Multimedia Analysis

We follow a statistical learning theory approach to tackle the high-level multimedia analysis problem. Figure 1.6 illustrates the proposed semantic-multimedia analysis algorithm. As can be seen in the diagram our work is built on top of the output of low-level multimedia analysis algorithms: semantics of multimedia information is represented as a statistical model of low-level feature data estimated from training data. Other approaches would also employ metadata and other sources of information.

In the first step we process a multimedia document by dividing it into text-image pairs, and then extract the low-level features from the different data types. In the second step we transform all feature spaces (text, colour, and texture) into a new data representation where keywords are easily modelled with an inexpensive and effective statistical model. These first two steps are described in Chapter 4. Finally, we represent a keyword as a linear model for its advantages for this task: support of high-dimensional data; ability to handle heterogeneous types of data; and low computationally cost. This last step is described in Chapter 5.

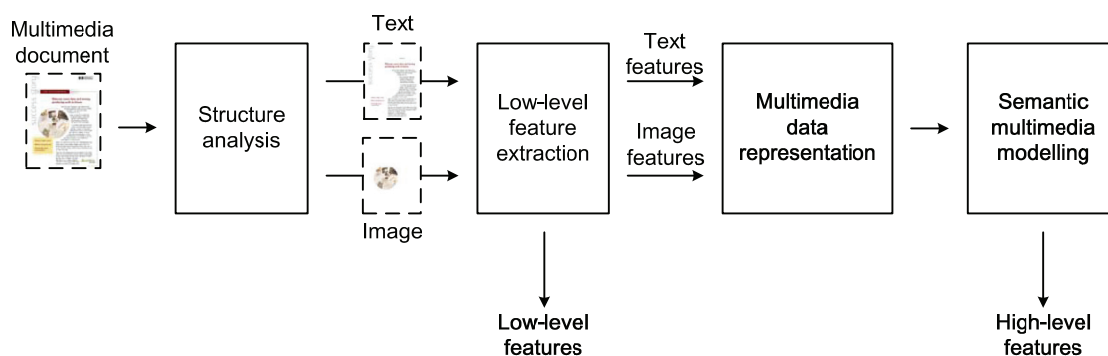


Figure 1.6. The semantic-multimedia analysis process.

1.4.2 Search-by-Keyword

The developed high-level analysis algorithm provides a set of keyword probabilities that enable multimedia information to be searched with a vocabulary of predefined keywords. The implemented search-by-keyword paradigm allows the user to submit a query with logical expression of keywords and corresponding weights. This produces one or more query vectors that are then used to search for the documents that are most similar to that query vector.

1.4.3 Search-by-Semantic-Example

The implemented search-by-semantic-example paradigm applies the high-level analysis on the query example to obtain the corresponding keyword probabilities. To find the documents that are most similar to the query vector we use the same strategy as for the previous case. Several examples can be provided and they are combined according to the logical expression submitted by the user. Moreover, both search-by-keyword and search-by-semantic-example can be used simultaneously to improve the expressiveness of the user information needs. Chapter 6 presents a framework to improve the user query expressiveness and investigates methods to compute the semantic similarity between the queries and the document vectors.

1.5 Contributions

The research carried out during the last few years resulted in an accumulated expertise that materialises in the following contributions to the scientific community:

1. A better understanding of the multimedia information retrieval research area with a published survey of semantic multimedia analysis algorithms, and a discussion of the problems of evaluating IR systems on semantic-multimedia documents (Chapters 2 and 3);
2. A practical and efficient method to build a high-dimensional visual vocabulary. This method

transforms low-dimensional feature spaces into high-dimensional feature spaces that allow to represent efficiently a vast number of concepts (Section 4.4);

3. An estimation of the size of the visual vocabulary that can be obtained by the minimum description length principle (Section 4.3);
4. A thorough study of linear algorithms as keyword models of semantic-multimedia: Rocchio classifier, naïve Bayes, and logistic regression with L_2 regularization (Sections 5.2 and Section 5.3);
5. Algorithms that have a low computational complexity and are semantically scalable (Subsections 5.6.3 and 5.6.4);
6. Proposed a keyword space to search semantic-multimedia by example (Section 7.2);
7. A characterization of the keyword space in terms of its similarity functions, dimensionality and the influence of semantic analysis accuracy (Section 7.6);
8. All developed software is available for download.

1.6 Publications

In this section we list the publications that disseminated the research results obtained with the research presented in this thesis. Publications are grouped by area of contribution.

Reviews

Previously published techniques to analyse multimedia information represent a vast expertise offering an excellent insight to the area. Thus, to better assimilate and organize the different techniques the following review was published and updated in Chapter 3:

- João Magalhães and Stefan Rüger, “Semantic multimedia information analysis for retrieval applications,” book chapter, Ed. Yu-Jin Zhang, “Semantic-based visual information retrieval,” IDEA group publishing, 2006.

Semantic-Visual Analysis

The initial work on modelling semantic information started with visual information with the idea of creating a generic codebook of visual words as a way of representing all possible visual information. Under this assumption keywords needed to be expressed with a combination of these visual words. To achieve this goal I implemented several algorithms:

- K2 algorithm: João Magalhães and Stefan Rüger, “Mining multimedia salient concepts for incremental information extraction,” poster at ACM SIGIR Conference on research and development in information retrieval, Salvador, Brazil, August 2005.
- Logistic regression: João Magalhães and Stefan Rüger, “Logistic regression of generic codebooks for semantic image retrieval,” International Conference on image and video retrieval, Phoenix, AZ, USA, July 2006.
- Naïve Bayes and Rocchio classifier: João Magalhães and Stefan Rüger, “High-dimensional visual vocabularies for image retrieval,” poster at ACM SIGIR Conference on research and development in information retrieval, Amsterdam, The Netherlands, July 2007.

Semantic-Multimedia Analysis

Modelling multimedia information was a required step to address the problem of semantic-multimedia information retrieval. Text was also processed into a codebook of text terms similarly to visual terms. This resulted in a completely automatic framework to analyse text documents, image documents, and text and image documents. This statistical framework was thoroughly investigated and published:

- João Magalhães and Stefan Rüger, “Information-theoretic semantic multimedia indexing,” ACM Conference on image and video retrieval, best paper award, The Netherlands, July 2007.
- João Magalhães and Stefan Rüger, “An information-theoretic framework for semantic-multimedia analysis,” *Journal article to be submitted*.

Searching Semantic-Multimedia

Once the semantic-multimedia analysis algorithms are in place, it becomes possible to exploit the semantics of multimedia documents in many different ways. Search-by-keyword and search-by-semantic-example are two search paradigms that were investigated and published:

- João Magalhães, Simon Overell and Stefan Rüger, “A semantic vector space for query by image example,” ACM SIGIR conference on research and development in information retrieval, Multimedia Information Retrieval Workshop, Amsterdam, The Netherlands, July 2007.
- João Magalhães, Fabio Ciravegna and Stefan Rüger, “Exploring multimedia in a keyword space,” ACM Multimedia, Vancouver, Canada, November 2008, *accepted for publication*.

- João Magalhães and Stefan Rüger, “Searching semantic-multimedia by example,” *Journal article to be submitted*.

1.7 Organization

The goal of this chapter was to present the general research semantic-multimedia information retrieval problem and to position each chapter and contribution of this thesis in its due place. Next, we present some background material:

- **Chapter 2 – Evaluation methodologies:** covers all aspects of information retrieval systems evaluation: information metrics, scalability metrics, and reference collections.

The first part of this thesis addresses the problem of semantic-multimedia indexing:

- **Chapter 3 – Semantic-multimedia analysis:** discusses several models for semantic-multimedia analysis – more emphasis is put on text and image analysis algorithms, and on automatic semantic search methods.
- **Chapter 4 – A multi-modal feature space:** details how we find an “optimal” representation of our multimodal data which is easily modelled by the family of statistical models used in Chapter 5.
- **Chapter 5 – Keyword models:** describes how a keyword is expressed as a statistical model of multi-modal data. The family of linear models is particularly adequate for this task for its support of high-dimensional data and ability to handle heterogeneous types of data.

The second part of this thesis addresses the problem of searching semantic-multimedia:

- **Chapter 6 – Searching multimedia:** discusses several methods of searching multimedia and discusses how semantic indexing creates a new search paradigm.
- **Chapter 7 – Keyword spaces:** proposes a search by semantic example paradigm. More specifically we study the different characteristics of a keyword space and compare automatic multimedia analysis methods to manual annotation.

Evaluation Methodologies

2.1 Introduction

The large number of variables affecting an IR system makes it very difficult to assess it with a unique measure. IR evaluation has been widely studied and it has shown to be extremely useful to compare different systems: information retrieval effectiveness metrics measure how well the system can satisfy the user information need, efficiency metrics measures the system responsiveness to the user query and the system's ability to cope with large scale situations.

Effectiveness and efficiency results produced by an evaluation methodology are widely affected by the data that is used to test the system: a dataset can contain information with different complexities that affect precision; the size of the data can also affect recall or precision (increase in class confusion), the quality of relevant/non-relevant annotations, or even the notion of relevant documents. Hence, novel evaluation methodologies are now being investigated to address scenarios where the notion relevant/non-relevant document has evolved into one where there are different levels of relevance or where there is a single relevant document, e.g., Web IR, semantic IR, multimedia, question-answering, expert discovery.

In this chapter we introduce the traditional metrics and resources used in the evaluation of IR systems: **effectiveness measures**, **efficiency measures** and **datasets**.

2.2 Effectiveness

In response to a search query, the system being evaluated retrieves a ranked list of documents ordered by relevance. The ideal IR system would return a rank list containing all relevant documents at the top followed by non-relevant documents. Unfortunately, it is common to have a mixture of relevant and non-relevant documents at the top of the ranked list. Thus, it is

fundamental to compare ranking algorithms with some measure of how effectively algorithms place relevant documents at the top of the list. The effectiveness measure can be obtained for each query or for a given set of queries, allowing the evaluation to be done on a “per-search” basis or a “per-run” basis. While the per-search evaluation assesses the retrieval effectiveness for a particular query, the per-run evaluation assesses the system’s mean performance over all single queries. It is particularly interesting to verify whether an algorithm performs regularly well across all queries or if it performs extremely well on some and extremely bad on others.

Before introducing retrieval effectiveness measures we will discuss the meaning of relevance and see how its different interpretations can result in different assessment metrics.

2.2.1 Defining Relevance

Relevance is the central concept of Information Retrieval. It has been widely studied in different areas as the extensive review presented by Mizzaro (1997) shows. Mizzaro claims that relevance is a complex concept involving different aspects: methodological foundations, different types of relevance, beyond-topical criteria adopted by users, modes of expression of the relevance judgment, dynamic nature of relevance, types of document representation, and agreement among different judges. In this discussion we leave some aspects aside and merge the remaining aspects into two practical facets that are important to the design of semantic-multimedia information retrieval: **types of relevance; incomplete and inconsistent relevance judgments.**

Several research areas have their own definition of relevance giving more emphasis to their specific objectives – IR aims at finding documents that *best* answers an information need, i.e. the most relevant documents for a particular user query. Information retrieval relies on datasets of documents whose relevance for a given query was judged by a human. Unfortunately, there is no universal definition of what a relevant document is: the notion of a relevant document is diffuse because the same document can have different meanings to different humans. This has been discussed by several researchers that noticed discrepancies between relevance judgments made by different annotators, see (Voorhees 1998) and (Volkmer, Thom and Tahaghoghi 2007). These discrepancies are more visible in large multimedia collections for two reasons: (1) multimedia information is not as concrete as textual information, thus more open to different interpretations and relevance judgments (types of relevance); (2) assessing the relevance of documents is an expensive task involving humans during long periods of time, thus collections with a large number of documents are only partially annotated: relevance judgments are incomplete and inconsistent.

Types of Relevance

Systems are evaluated on collections of documents that were manually annotated by human assessors. According to the information domain, different definitions of relevance are more

adequate than others. We have identified three types of relevance that are valuable to evaluate multimedia information retrieval:

- **Binary relevance:** under this model a document is either relevant or not. It makes the simple assumption that relevant documents contain the same amount of information value. This approximation results in robust systems that achieve similar accuracy across different queries types, (Buckley and Voorhees 2000).
- **Multi-level relevance:** one knows that documents contain information with different importance for the same query, thus, a discrete model of relevance (e.g., relevant, highly-relevant, not-relevant) enables systems to rank documents by their relative importance. This type of relevance judgments allows assessors to rate documents with different levels of relevance for a particular topic.
- **Ranked relevance:** when documents are ordered according to a particular notion of similarity. An example of this type of relevance is when studying different image compression techniques users are asked to order compressed images by their quality in relation to the original.

The binary relevance model is a good reference to develop IR systems that serve a wide variety of non-specialized IR applications – the system is tuned with a set of relevance judgments that reflect the majority of human assessors’ judgments. Voorhees (2001) has showed empirically that systems based on binary relevance judgments are more robust and stable than the ones based on multi-level relevance judgments. This happens because in the second case, systems use a fine-grain model to create a rank with N groups corresponding to the different level of relevance. The ranking algorithm has the task of placing each one of the M documents in the correct group of relevance level. It is easy to see that this task is much more difficult and tuning such algorithms will easily lead to an overfitting situation that is less general, and therefore less robust and stable (Voorhees 2001).

The relevance judgments of the ranked relevance model are actually a rank of documents that exemplify the human perception of a particular type of similarity, e.g., texture, colour. The similarity function expressed by the rank is the ranking algorithm that is approximate. For this reason, these systems (and the evaluation metrics) are more stable and less prone to overfit than multi-level relevance systems. A disadvantage of this ranked relevance is the exponentially increasing cost of generating the ranked relevance judgments.

Incomplete and Inconsistent Relevance Judgements

Another practical problem concerning relevance in very-large scale collections is the incompleteness and inconsistency of relevance judgments. In some situations the evaluation collection is so large that human assessors cannot judge all possible documents (incomplete relevance judgments), and sometimes different annotators give different relevance judgement to the same document (inconsistent relevance judgments). These trends have been extensively studied by Voorhees (1998) and Buckley and Voorhees (2004) who proposed a metric to reduce the effect of incomplete relevance judgments. More recently Aslam and Yilmaz, presented more stable metrics in (Yilmaz and Aslam 2006; Aslam and Yilmaz 2007) to tackle the stability of measures under these conditions (incomplete and inconsistent relevance judgments).

One of the most important studies of human relevance judgments of multimedia information is the one presented by Volkmer, Thom, and Tahaghoghi (2007). They describe and analyse the annotation efforts made by TRECVID participants that generated the relevance judgments of all training data for 39 concepts of the high-level feature extraction. To overcome the problems of incomplete and inconsistent relevance judgments the following rules were followed:

1. Assessors annotated a sub-set of the documents with a sub-set of the concepts; this avoids the bias caused by having the same person annotating all data with the same concept.
2. All documents must receive a relevance judgment from all annotators; this eliminates the problem of incomplete relevance judgments but increases inconsistency.
3. Documents and concepts were assigned to annotators so that some documents received more than one relevance judgment for the same concept; this eliminates the inconsistency problem if a voting scheme is used to decide between relevant and non-relevant.

We stress the fact that this annotation effort was done on training data that is usually much larger than test data. So, the same problems of incomplete and inconsistent relevance judgments exist when systems are evaluated. This large scale effort was highly valuable for two reasons: it produced high-quality annotations of training data; and it gave important information on how humans judge multimedia information for particular queries, see (Volkmer, Thom and Tahaghoghi 2007) for more details.

2.2.2 Precision and Recall

Precision and recall are the two most popular metrics in information retrieval. These measures are applied on ranked lists with both relevant documents – marked as ‘+’ in Figure 2.1 – and non-relevant documents – marked as ‘-’ in Figure 2.1 – for the given query. The two metrics assess

different aspects of a system: precision addresses the accuracy of the system and recall addresses the completeness of the system.

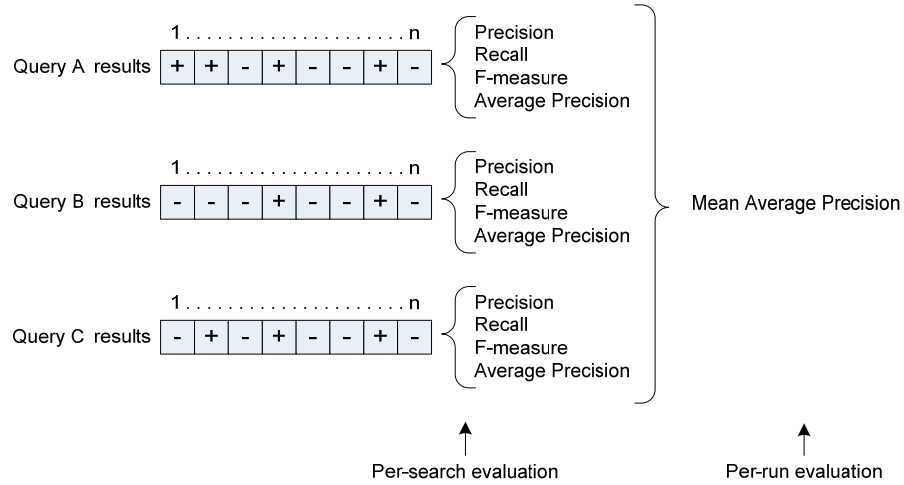


Figure 2.1. Retrieval effectiveness metrics based on relevant documents.

- **Precision (Prec):** a measure of the ability of a system to present only relevant items. The precision metric is expressed as

$$\text{Prec} = \frac{\text{relevant in first } n \text{ documents}}{n}. \quad (2.1)$$

- **Recall (Rec):** a measure of the ability of a system to present all relevant items. The recall metric is expressed as

$$\text{Rec} = \frac{\text{relevant in first } n \text{ documents}}{\text{total relevant}}. \quad (2.2)$$

- **F-measure (Harmonic mean):** the harmonic mean assesses the trade-off between precision and recall. The F-measure is expressed as

$$F = \frac{2}{\frac{1}{\text{Prec}} + \frac{1}{\text{Rec}}}. \quad (2.3)$$

Each system should tune the retrieval model to improve the most relevant measure to the systems application, e.g., a patent information retrieval system should not miss any relevant document – this corresponds to a high recall system. Precision-recall curves are another useful way of visualizing a system’s retrieval effectiveness in detail. Figure 2.2 presents the examples of three systems. These curves are obtained by plotting the evolution of the precision and recall measures

along the retrieved rank. An ideal system would achieve both 100% precision and 100% recall. In practice systems always have a trade-off between precision and recall.

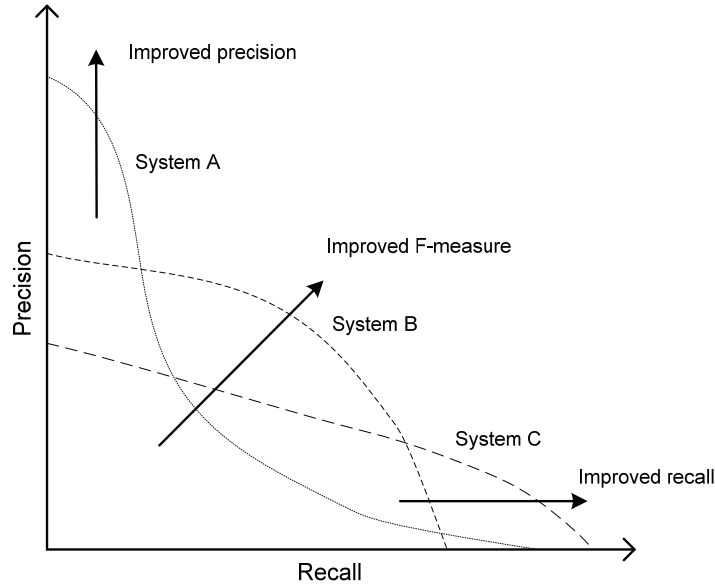


Figure 2.2. Interpretation of precision-recall curves.

A measure that gives more emphasis to relevant documents retrieved at the top of the rank is the Average Precision:

- **Average Precision (AP):** the average of the precision scores obtained after each relevant document is retrieved. Assuming that k relevant documents were retrieved, the average precision expression is:

$$AP = \frac{\sum_{k \in \{r | r \text{ is rank of relevant docs}\}} \text{Prec}@k}{|\text{Relevant docs}|}. \quad (2.4)$$

The previous measures evaluate the performance of retrieval results for a given single keyword. Assessing the retrieval effectiveness of a given system is done across several different query topics with a well known metric:

- **Mean Average Precision (MAP):** this metric summarizes the overall system retrieval effectiveness into a single value as the mean of all keywords' average precision,

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP_q. \quad (2.5)$$

Buckley and Voorhees (2000) have studied the required number $|Q|$ of different query topics to obtain statistically significant measures which allows to compare systems. They confirmed previous results (Voorhees 1998) suggesting that “at least 25 and 50 is better”, moreover, under the multi-level relevance model a minimum of 50 different query topics is required to obtain stable measures.

2.2.3 Metrics Generalization and Normalization

The above metrics give an indication of the effectiveness of an algorithm in a fixed evaluation scenario. The obtained measures are only valid for that specific scenario and cannot be generalized to other situations. Huijsmans and Sebe (2005) discuss how precision and recall based measures have a limited scope because they do not consider the number of relevant documents and the amount of noise in the collection. Thus, the value provided by the metric does not generalize to other collections because it is not normalized by the information complexity of the collection. In Section 2.5.2 we discuss the information complexity of a collection.

2.3 Efficiency

When discussing efficiency of an IR system, one can address different functions of the complete system. In semantic-multimedia IR we are interested in the extra computational complexity over conventional multimedia IR systems. This extra complexity resides in the extra processing required to execute the analysis algorithms to extract the semantics of multimedia content and in the computation of the models that enable the analysis algorithms.

Similarly to traditional IR the development of the model is done offline and tuned in a lab to best fit the training data. Thus, the learning complexity is only relevant if one is in the presence of a relevance feedback system, which we do not address in this thesis. Therefore, we solely focus on the runtime complexity of the analysis algorithms.

2.3.1 Indexing Complexity

The indexing of information as discussed in Chapter 1 involves the generation of indexing tokens and its storage in a way that it can be efficiently accessible. The added complexity to index semantic-multimedia corresponds to the semantic-multimedia analysis algorithm. More specifically we are interested in:

- **Time complexity:** how many documents/concepts per second can an algorithm process – time complexity is a variable that affects the system responsiveness;

- **Space complexity:** the memory required to process a document for the entire vocabulary – space complexity is variable that affects the system scalability;

While time complexity defines the minimum time in which a request can be satisfied, the second defines how well a system scales with several simultaneous requests. There is a trade-off between the above two variables.

2.3.2 Query Analysis Complexity

The query analysis complexity corresponds to the cost of processing the standard query-parsing methods as in traditional IR systems added by the cost of running the semantic-multimedia analysis algorithms. Note that the last cost only exists for query-by-example queries, other queries do not incur additional costs. Thus, the additional cost for query-by-example is equivalent to cost of running the algorithm on the example provided by the user (it is equivalent to the extra indexing cost).

2.4 Collections

Evaluation measures are not the only tools involved in assessing semantic-multimedia information retrieval systems – multimedia collections also play an important role. Multimedia collections are research tools that provide a common test environment to evaluate and compare different algorithms. Collections exist to evaluate many different algorithms such as shot-boundary detection, low-level visual features, story segmentation, keyword based retrieval or automatic and semi-automatic search. This thesis addresses the problem of indexing and searching multimedia by its semantic content. Thus, the two following aspects are required to be present in our collections:

- **Keywords** corresponding to concepts present in the collection content are used to describe which meaningful concepts are present in individual multimedia documents.
- **Categories** are groups of multimedia documents whose content concern a common meaningful theme, i.e., documents in the same category are semantically similar.

The above definitions create two types of content annotations – at the document level (keywords) and at the group of documents level (categories). While the first set of annotations is used to develop and evaluate the semantic-multimedia analysis algorithms (Chapter 5), the second set of annotations corresponds to the queries on the evaluation of the semantic-multimedia search evaluation (Chapter 7). Table 2.1 summarizes all collections used in this thesis. We shall describe next these collections in detail and present and discuss other related collections.

Collection	Images	Text	Training	Test	Keywords	Categories
Reuters-21578		✓	7,770	3,299	90	0
Corel5000	✓		4,500	500	179	50
TRECVID ⁵	✓	✓	23,709	12,054	39	8

Table 2.1. Summary of evaluation collections used in this thesis.

2.4.1 Text Collections

Both retrieval and semantic description types of measures will be presented in the following sections, followed by a discussion of some of the image and video datasets that are commonly used in semantic-multimedia information retrieval applications.

Reuters-21578

This is a widely used text dataset, which allows comparing our results with others in the literature. Each document is composed by a text corpus, a title (which we ignore), and labelled categories. This dataset has several possible splits, and we used the predefined *ModApte* split for the keyword models evaluation:

- **Keyword models evaluation:** the *ModApte* split contains 9,603 training documents and 3,299 test documents, and it has been used in several other publications (Joachims 1998; Nigam, Lafferty and McCallum 1999; McCallum and Nigam 1998; Zhang and Oles 2001). Terms appearing less than 3 times were removed. Only labels with at least 1 document in the training set and the test set were considered leaving us with 90 labels. After these steps we ended with 7,770 labelled documents for training.

20-Newsgroup

This collection is made up of newsgroups posts of different users to 20 newsgroups concerning different topics of discussion. The collection contains approximately 20,000 posts and it is evenly split (10,000 training and 10,000 test). The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques such as text classification and text clustering.

2.4.2 Image Collections

Several image datasets exist on the web or have been made available by researchers. We will discuss the image datasets that were used in the publications of this thesis.

⁵ This collection corresponds to a random split of the TRECVID development data.

Caltech 256

Caltech256 is collection of 30,608 images that was carefully created for object recognition. It contains 256 object categories and an extra category of noise. The collection was designed with the goal of obtaining a high-quality set of training data: a taxonomy of objects was created to cover a diverse and realistic collection; all object categories have enough examples (>80) to train a statistical model and reduce the effect of overfitting; images were collected from the Web with Google and PicSearch; only 30% of the total 92,652 collected images were deemed as “usable”.

Corel Images: Corel Stock Photo CDs

Corel Stock Photo CDs are a compilation of professional photographs organised by topics (e.g., Arabian horses, cars races, sunsets) with different resolutions and manually annotated with keywords. The images in each CD belong to the same category. It is known that the way in which images are selected from the CDs to build the training and testing set greatly influences retrieval performances (Müller, Marchand-Maillet and Pun 2002).

In the past the Corel photo collection has been criticised for being an easy collection from an image retrieval point of view, see (Westerveld and de Vries 2003a). Nevertheless, the subset created by Duygulu et al. (2002) is a very popular dataset that is a reference dataset to compare different retrieval algorithms. It consists of 5,000 image, 4,500 for training and 500 for testing, and each image has 1-5 keywords from a vocabulary of 371 words. This collection is used in two types of experiments:

- **Keyword model evaluation:** only keywords with at least 2 images in the test and training set each were considered which reduces the size of the vocabulary to 179 keywords. Keyword models were learned on the training set and were evaluated on the test set.
- **Semantic search evaluation:** the collection is already organized into 50 image categories, such as rural France, Galapagos wildlife and nesting birds. The semantic categories are used as semantic-query topics for evaluation and only the test set is used in this evaluation.

Getty Images

The Getty Images (<http://creative.gettyimages.com>) dataset compiled by Yavlinsky et al. (2005) is a selection of photographs obtained by submitting queries, which exclude any non-photographic content, any digitally composed or enhanced photos and any photos taken in unrealistic studio settings. The resulting dataset contains pictures from a number of different photo vendors, which reduces the chance of unrealistic correlations between keywords and image contents. Keywords for Getty images come in three different flavours: subjects (e.g., *tiger*), concepts (e.g., *emptiness*) and styles (e.g., *panoramic photograph*).

ImageCLEF2007: Flickr Images and Descriptions

The ImageCLEF Photo collection includes 20,000 images and corresponding metadata (both textual descriptions and geographic information). The photos vary in quality, levels of noise, and illustrate several concepts, actions or events. Metadata enriches the images by adding information such as the fact that a street is in some location or the profession of one of the persons in the photos.


	DOCNO	annotations/00/60.eng
	TITLE	Palma
	NOTES	The main shopping street in Paraguay
	LOCATION	Asunción, Paraguay
	DATE	March 2002
	IMAGE	images/00/60.jpg
	THUMBNAIL	thumbnails/00/60.jpg

Figure 2.3. Example of an ImageCLEF document.

Figure 2.3 illustrates an example of metadata information on the ImageCLEF2007 collection. The goal of this dataset is to simulate a scenario where collections have heterogeneous sources of data and users submit textual queries together with visual examples. A more thorough description of the dataset can be found in (Grubinger et al. 2007).

2.4.3 Video Collections

In this section we discuss the underlying patterns of different video genres and describe a video collection that covers the discussed genres.

TV News and Documentaries

TV news and documentaries are an essentially informative type of content. There is a story that a reporter or a commentator is telling and the images are used to illustrate that story. Consequently, to extract the meaning of the content of TV news and documentaries programs, algorithms must be centred on the speech (or the ASR text), while the audio and visual parts can be used as auxiliary data. However, if one wishes to detect the presence of a given concept on the video programme, then its analysis must be centred on the visual modality.

Narratives: Movies and Soap Operas

This type of content contrasts with the previous type in the sense that it is the video that tells the story or narrative. In a movie or soap opera, the speech track usually consists of the characters' lines, and the sound and the visual tracks tell another part of the narrative as a complement to the characters' lines. Many directors choose to use only sound and visual media for scenes where they

want to transmit stronger emotions, e.g., the legendary shower curtain scene in Hitchcock's *Psycho*.

Movies and soap opera directors follow a set of production rules, which result in a general structure that can be modelled by some statistical tools, as was shown by Sundaram and Chang (2000) and Vasconcelos and Lippman (2000) who analysed a set of Hollywood movies.

Sports Videos

Sports videos are a peculiar type of video content because the constrained structure of the video, composed by a small number of standard shots, makes it an easier type of content to analyse than the previous cases. In all sports, there are a set of pre-defined events and states (due to the rules of the game) that follow a given pattern or sequence. The semantics are therefore reduced by a given set of constraints. As a consequence, in sports videos the semantic ambiguity is much lower than in other type of videos.

TRECVID

The growing interest in managing multimedia collections effectively and efficiently has created a new research interest arising as a combination of multimedia understanding, information extraction, information retrieval and digital libraries. This growing interest has initially resulted in the creation of a video retrieval track in the TREC conference series which later developed into a workshop in its own right.

To run our experiments on video data we used the TRECVID data: since only the training set is completely labelled, we randomly split the English training videos into 23,709 training documents and 12,054 test documents. We considered each document to be a key-frame plus the ASR text within a window of 6 seconds around that key-frame. Thus, this collection of key-frames documents were used in the two following ways:

- **Keyword model evaluation:** We evaluated the keyword models on the 39 keywords of the standard vocabulary provided by NIST. The training set was used to learn the keyword models and the test set was used to test the corresponding models.
- **Semantic search evaluation:** The 8 categories were selected from the large-scale LS-COMM ontology of 400 keywords provided by Naphade et al. (2006). We selected those 8 categories as non overlapping keywords with the other 39 keywords and had a large enough number of examples. Only the test set is used in this evaluation.

2.5 Collection Generation

The generation of collections for evaluations is a task that must be considered and designed

carefully to adequately reflect and preserve the problem characteristics. In the course of this thesis we acquired some experience from working with several collections and we deemed the following four aspects to be crucial in the generation of such collections: data sampling strategies, data complexity, information relevance and generalization and cross-datasets.

2.5.1 Data Sampling Strategies

The generation of collections is influenced by the sources from where data is gathered and by the heuristics used to select the relevant examples from those sources. Sampling strategies ought to correctly reflect the problem in terms of the number of classes, examples per class, and ideally samples should be independent and identically distributed for each class. This is not always possible because every method used to gather information will do it in a particular way that introduces a natural bias.

2.5.2 Data Complexity

The complexity of a problem is generally reflected in the structure of its data. Data complexity greatly affects the number of samples required to estimate a particular statistical model. Entropy based measures can be used to infer the relative complexity of a particular set of data and force the sampling strategy to gather more data for that particular case. Thus, balanced sampling should be modified to gather more samples of specific classes if required.

2.5.3 Information Relevance

Assessing the user relevance is always a problem with some ambiguity inherent to the nature of the handled information. For example, news about a movie and its profits might fall into the *entertainment* or *business* news categories. As we discussed in Section 2.2.1, relevance is a highly subjective concept and in multimedia IR it becomes a critical problem. This is rooted in three main issues:

- Collections are sometimes too generic
- Precision based measures might not reflect the user satisfaction
- Algorithms are too generic and lack the focus to answer the user's subjective needs

These issues become more critical when one considers semantic-multimedia information. In these cases, retrieval is ambiguous due to the nature of the information: there are no explicit symbols in data; it relies on the human knowledge and interpretation of the data under a certain

query and the corresponding context. Therefore, it is critical that researchers fully understand the scenario and knows how to address the user information needs, i.e., the problem classes (ontology or taxonomy), and its corresponding examples need to be carefully identified.

2.5.4 Generalization and Cross-Datasets

The generalization of a learning method relates to its capabilities to extract information from semantic-multimedia content. The assessment of this characteristic is quite important as the content will be made accessible according to the output of those algorithms. In semantic-multimedia IR, generalization takes an extra meaning: because we aim at creating models of concepts by learning it on particular training data, we wish to detect that concept on test data coming from any source. Unfortunately, the natural bias of the training collection is always present and if one tests it on a different type of collection it will not be as effective. For instance, examples of *cars* on the Corel collection are quite different from examples of cars on the TRECVID collection. Thus, a model trained on the TRECVID collection will not be as effective on the Corel collection as it is on the TRECVID.

Yavlinsky and R uger (2007) suggested to model a keyword on a given dataset with keywords models of other dataset. Moreover, in order to test how well a concept generalizes they suggested estimating keywords on a given collection (obtained with a given sampling strategy) and test it on another collection. This cross-collection evaluation can give a good indication as to how well the learning algorithm is capable of generalizing to data from other sources but referring to same reality.

2.6 Summary

The assessment and comparison of different semantic-multimedia IR algorithms requires common evaluation scenarios with well defined evaluation measures and reference collections. While evaluation measures are an objective and clear part of the process, multimedia collections are not so clear and they do affect the evaluation of algorithms. Depending on the collection characteristics, e.g., media-type, genre, artistic features, some analysis algorithms are more adequate than others. Each collection has different information complexities with different abstraction levels or artistic characteristics, thus, establishing a specific level of difficulty associated to a particular collection. To compare different algorithms one must use the same collection and the same evaluation methodology.

Part 1
Indexing Semantic-Multimedia

Semantic-Multimedia Analysis

3.1 Introduction

Multimedia information retrieval is a research area that brings together many different expertises required for each specific problem. In the first part of this thesis we shall focus on the analysis of semantic-multimedia for indexing. Chapter 3 presents a literature review, and Chapters 4 and 5 propose a novel semantic-multimedia analysis framework. The main goal is to infer the semantics of multimedia information so that it can be easily searched by its semantic content. Figure 3.1 illustrates the scope of the first part (the dashed blocks are addressed in the second part).

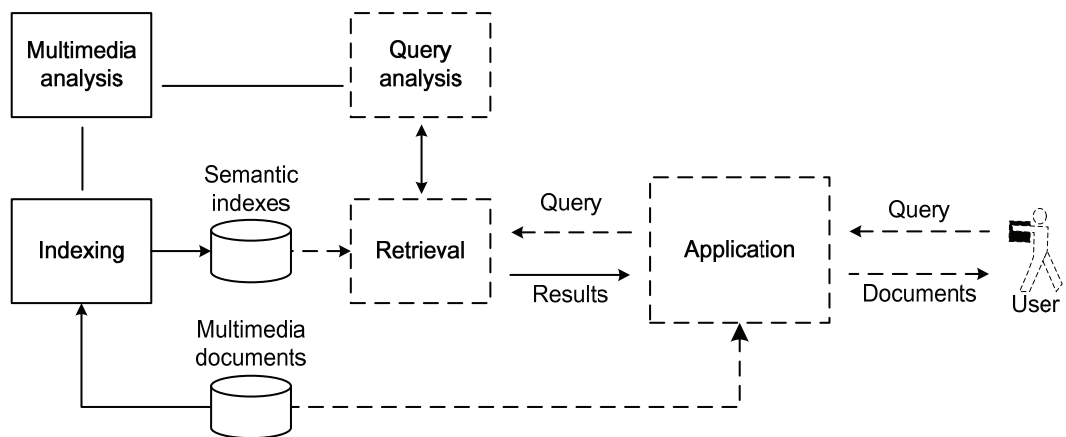


Figure 3.1. Scope of the semantic-multimedia analysis problem.

Algorithms that extract the semantics of multimedia content are dependent on the content media type (e.g., visual, text, audio, sound), on the type of approach (e.g., statistical, rule based) and on the type of input data (e.g., metadata, low-level features) used to infer semantics. The review presented in this Chapter is an extended and updated version of the survey published in (Magalhães and Rüger 2006) and it is organized into single-media and multi-modal content analysis algorithms.

3.2 Single-Media Analysis

This section will discuss analysis algorithms for images, text and audio. The goal of the presented algorithms is the creation of a model per keyword (either statistical or rule based) that enables a system to extract information from that media type.

3.2.1 Image Analysis

Image analysis and understanding is one of the oldest fields in pattern recognition and artificial intelligence. A lot of research has been done since Marr (1983), culminating in the modern reference texts by Forsyth and Ponce (2003), and Hartley and Zisserman (2004). Datta et al. (2008) have recently presented a survey offering a thorough and insightful look at the area of image retrieval. In this section we shall discuss several algorithms according to their type:

- **Single class models** fit a simple probability density distribution to each keyword
- **Translation models** defines an intermediate representation of visual features and a method to translate from this representation to keywords
- **Hierarchical and network models** explore the inter-dependence of image elements (regions or tiles) and its structure
- **Knowledge based models** improve the models' accuracy by including other sources of knowledge besides the training data, e.g., a linguistic database, WordNet

Single Class Models

A direct approach to the semantic analysis of multimedia is to learn a class-conditional probability distribution $p(w | d)$ of each single keyword w of the semantic vocabulary given its training data d , see Figure 3.2. This distribution can be obtained using Bayes' law

$$p(w | d) = \frac{p(w)p(d | w)}{p(d)}. \quad (3.1)$$

The keyword probability $p(w)$ can be computed straightforward and the $p(d | w)$ can be computed with very different data density distribution models $p(d)$. Several techniques to model $p(d | w)$ with a simple density distribution have been proposed: Yavlinsky et al. (2005) used a nonparametric distribution, Carneiro and Vasconcelos (2005) a semi-parametric density estimation, Westerveld and de Vries (2003b) a finite-mixture of Gaussians, and Mori et al. (1999), Vailaya et al. (1999), Vailaya et al. (2001) different flavours of vector quantization techniques.

Wang et al. (2001) and Yavlinsky et al. (2005) modelled $p(d | w)$, the probability density of images given keywords, as a nonparametric density smoothed by two different kernels: a Gaussian kernel and an Earth Movers Distance kernel. They used both global and 3 by 3 tile colour features and texture features. The best reported mean average precision (MAP) results with tiles achieved 28.6% MAP with the dataset of Duygulu et al. (2002) and 9.2% with a Getty Images dataset.

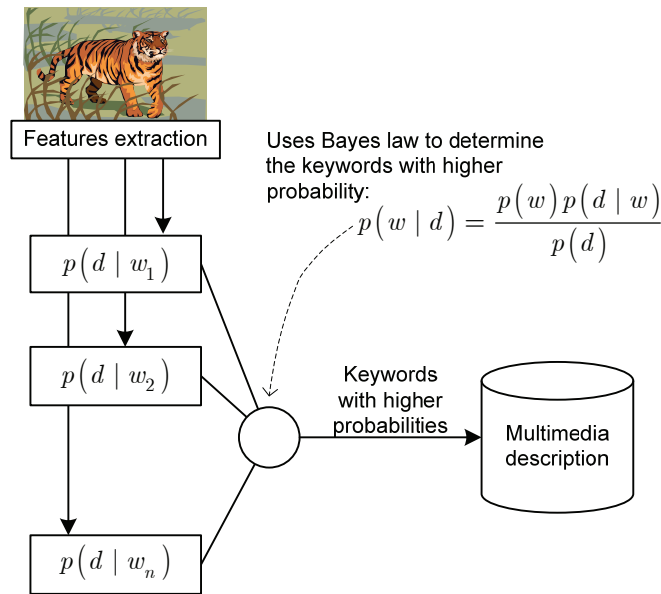


Figure 3.2. Inference of single class models.

Yavlinsky et al. (2005) showed that a simple nonparametric statistical distribution can perform as well as or better than many other sophisticated techniques, e.g., translation models. However, the nonparametric density nature of their framework makes the task of running the model on new data very complex. The model is the entire dataset meaning that the demands on CPU and memory increase with the training data.

Westerveld and de Vries (2003b) used a finite-mixture density distribution with a fixed number of components to model a subset of the DCT coefficients:

$$p(x | \theta) = \sum_{m=1}^k \alpha_m p(x | \mu_m, \sigma_m^2), \quad (3.2)$$

where k is the number of components, θ represents the complete set of model parameters with mean μ_m , covariance σ_m^2 , and component prior α_m . The component priors have the constraints $\alpha_1, \dots, \alpha_k \geq 0$ and $\sum_{m=1}^k \alpha_m = 1$. Westerveld and de Vries (2003b) tested several scenarios to evaluate the effect (a) of the number of mixture components, (b) of using different number of DCT coefficients (luminance and chrominance), and (c) of adding the coordinates of the DCT

coefficients to the feature vectors. The two first factors produced varying results, and optimal points were found experimentally. The third tested aspect, the inclusion of the coefficients' coordinates, did not affect results.

Combining the two previous approaches, Carneiro and Vasconcelos (2005) deployed a hierarchy of semi-parametric mixtures to model $p(x | w)$ using a subset of the DCT coefficients as low-level features. Vasconcelos and Lippman (2000) had already examined the same framework in a content-based retrieval system. The hierarchy of mixtures proposed by Vasconcelos and Lippman (1998) can model data at different levels of granularity with a finite mixture of Gaussians. At each hierarchical level l the number of each mixture component k^l differs by one from adjacent levels. The hierarchy of mixtures is expressed as

$$p(x | w_i) = \frac{1}{D} \sum_{m=1}^{k^l} \alpha_{i,m}^l p(x | \theta_{i,m}^l). \quad (3.3)$$

The level $l = 1$ corresponds to the coarsest characterization. The more detailed hierarchy level consists of a nonparametric distribution with a kernel placed on top of each sample. The only restriction on the model is that if node m of level $l + 1$ is a child of node n of level l , then they are both children of node p of level $l - 1$. The EM algorithm computes the mixture parameters at level l given the knowledge of the parameters at level $l + 1$, forcing the previous restriction. Carneiro and Vasconcelos (2005) report the best published retrieval MAP of 31% with the dataset of Duygulu et al. (2002).

Even though the approaches by Carneiro and Vasconcelos (2005) and Westerveld and de Vries (2003b) are similar, the differences make it difficult to carry out a fair comparison. The DCT features are used in a different way, and the semi-parametric hierarchy of mixtures can model keywords with few training examples.

The relationship between finite-mixtures density modelling and vector quantization is a well studied subject, see (Hastie, Tibshirani and Friedman 2001). One of the applications of vector quantization to image retrieval and annotation was deployed by Mori et al. (1999). Given the training data of a keyword, they divide the images into tiles and apply vector quantization to the image tiles to extract the codebook used to estimate the $p(d | w)$ density distribution. Later, they use a model of *word co-occurrence* on the image tiles to label the image. The words with the higher sum of probabilities across the different tiles are the ones assigned to that image.

Vailaya et al. (1999) and (2001) describe a Bayesian framework with a codebook to estimate the density distribution of each keyword. They show that the minimum description length criterion selects the optimal size of the codebook extracted from the vector quantiser. The features are extracted from the global image, and there is no image tiling. The use of the MDL criterion makes

this framework quite elegant and defines a statistical criterion to select every model parameter without any user-defined parameters.

Maximum entropy techniques have also been successfully applied to a number of language tasks such as speech recognition. Jeon and Manmatha (2004) deploy a maximum entropy framework to capture the relationships between words and the codebook of image tiles. They used the same discrete codebook as in their first work (Jeon, Lavrenko and Manmatha 2003) and report better results. One might expect that better results can be achieved after using a soft-codebook and using a different distribution for the image keywords.

Translation Models

All the previous approaches employ a model to directly estimate $p(d | w)$ in terms of low-level image features. In contrast, translation models generate an intermediate representation of images (e.g., a visual vocabulary) and express keywords in terms of the auxiliary representation, see Figure 3.3. The problem is equivalent to cross-language problems involving three languages, e.g., from Arabic to French and then to English. Four methods of creating intermediate representations have been studied: unsupervised computation of intermediate representation, linear decomposition of the co-occurrence matrix (latent semantic analysis, LSA), probabilistic approximation to the full matrix decomposition (probabilistic LSA) and a Bayesian approach to pLSA with a latent Dirichlet prior over the co-occurrence random variables (latent Dirichlet allocation, LDA).

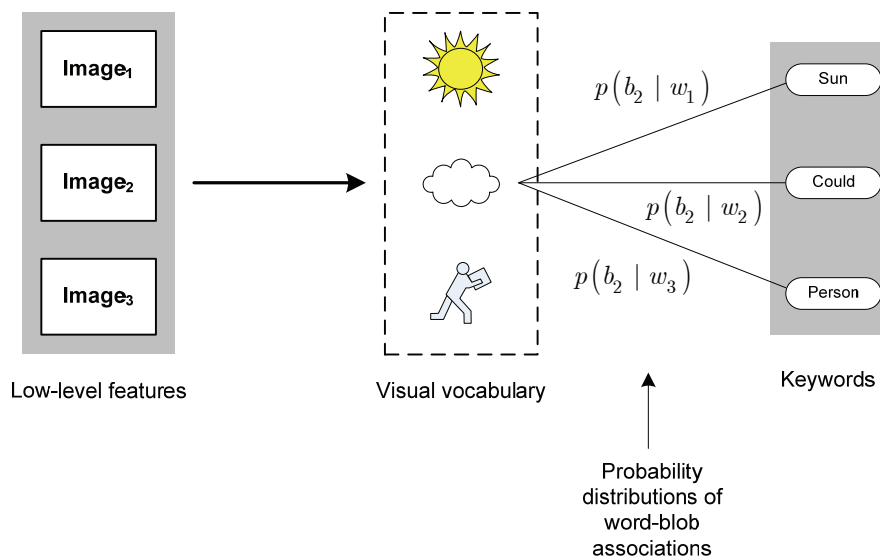


Figure 3.3. Translation models.

Inspired by machine translation research, Duygulu et al. (2002) developed a method of annotating image regions with words. First, regions are created using a segmentation algorithm like normalised cuts (Shi and Malik 2000). For each region, features are computed and then blobs are

generated by clustering the regional image features across an image collection. The problem is then formulated as learning the correspondence between the discrete vocabulary of blobs and the image keywords. The model consists of a mixture of correspondences between each word and each image in the collection,

$$p(w | b) \propto \sum_{i \in \{\text{blobs in } I_n\}} p(a_{nj} = i) p(w = w_{nj} | b = b_{ni}), \quad (3.4)$$

where $p(a_{nj} = i)$ expresses the probability of assigning the j^{th} word to blob i in image n , and $p(w = w_{nj} | b = b_{ni})$ is the probability of obtaining an instance of word w_j given blob b_{ni} . These two probability distributions are estimated with the EM algorithm. The authors refined the lexicon by clustering indistinguishable words and ignoring the words with probabilities $p(w | b)$ below a given threshold. The machine translation approach, the thorough experiments and the dataset form strong points of Duygulu's et al. (2002) contribution. This dataset is nowadays a reference, and thorough experiments showed that (a) their method could predict numerous words with high accuracy, (b) increasing the probability threshold improved precision but reduced recall and (c) the words clustering improved recall and precision.

Following a translation model Jeon, Lavrenko and Manmatha (2003), Lavrenko, Manmatha and Jeon (2003), and Feng, Lavrenko and Manmatha (2004) studied a model where blob features $b_I^{(r)}$ of an image I are assumed to be conditionally independent of keywords w_i , i.e.,

$$\begin{aligned} p(w_i, b_I) &= \sum_{J \in D} p(J) p(w_i | J) p(b_I | J) \\ &= \sum_{J \in D} p(J) p(w_i | J) \prod_{r \in I} p(b_I^{(r)} | J). \end{aligned} \quad (3.5)$$

Note that $b_I^{(r)}$ and w_i are conditionally independent given the image collection D and that $J \in D$ act as the hidden variables that generated the two distinct representations of the same process (the words and the features). Jeon, Lavrenko and Manmatha (2003), recast the image annotation as a cross-lingual information retrieval problem applying a cross-media relevance model based on a discrete codebook of regions. Lavrenko, Manmatha and Jeon (2003) continued their previous work (Jeon, Lavrenko and Manmatha 2003) and used continuous probability density functions $P(b_I^{(r)} | J)$ to describe the process of generating blob features and to avoid the loss of information related to the generation of the codebook. Extending their previous work, Feng et al. (2004) replace blobs with tiles and model image keywords with a Bernoulli distribution. This last work reports their best results, a MAP of 30%, with a Corel dataset (Duygulu et al. 2002).

There are a vast number of approaches inspired by latent semantic analysis (LSA), a technique

proposed in the early nineties (Deerwester et al. 1990), that exploits word/features co-occurrence to reduce the data space to a canonical space representation. LSA looks at patterns of word distributions (specifically, word co-occurrence) across a set of documents. A matrix M of word occurrences in documents is filled with each word frequency in each document. The singular value decomposition (SVD) of matrix M results in the transformation to a singular space where projected documents can be efficiently compared. Zhao and Grosky (2003) transform low-level features into keywords with LSA. Colour features are taken from images and arranged in a bi-dimensional histogram (hue-saturation) with 100 bins in total. This histogram plays the same role as words in text. The matrix M is then filled with each row corresponding to a histogram of a given image. The SVD of this matrix is then computed to obtain the singular space. To analyse new images the SVD transformation matrices map new images to that space, and a nearest-neighbour technique selects the nearest keywords. This method has the advantage that it does not need a lot of training data for each keyword. More recently, Hare et al. (2006) proposed a linear-algebraic LSA based approach that creates a semantic space of low-level features and keywords. By applying SVD the correlations between features and keywords are reduced to a simpler form. The rank of the decompositions matrices is determined empirically. A thorough analysis of this method has been presented in (Hare, Samangooei and Lewis 2008).

A probabilistic formalization of LSA, pLSA (Hofmann 1999), offers a fresh view of LSA. pLSA is a statistical framework that approximates the SVD by a learning algorithm that jointly estimates the intermediate representation and the keyword correspondences. It does not define the correspondence learning algorithm itself, thus, a study of three different algorithms to learn the topic-keyword correspondence has been done in (Monay and Gatica-Perez 2007). Barnard and Forsyth (2001) studied a generative *hierarchical aspect model*, which was inspired by Hofmann and Puzicha's (1998) hierarchical clustering/aspect model. In this case the intermediate representation is hierarchical. The data are assumed to be generated by a fixed hierarchy of nodes, where the leaves of the hierarchy correspond to soft clusters. Mathematically, the process for generating the set of observations O associated with an image I can be described by

$$p(O | I) = \sum_c p(c) \prod_{o \in O} \left(\sum_l p(o | l, c) p(l | c, I) \right), \quad (3.6)$$

$$O = \{w_1, \dots, w_n, b_1, \dots, b_m\},$$

where c indexes the clusters, o indexes words and blobs, and l indexes the levels of the hierarchy. The level and the cluster uniquely specify a node of the hierarchy. Hence, the probability of an observation $p(o | l, c)$ is conditionally independent given a node in the tree. In the case of words

$p(o | l, c)$ assumes a tabular form, and in the case of blobs a Gaussian models the regions' features. As in the original pLSA, the model is estimated with an EM algorithm.

Blei and Jordan (2003) describe three hierarchical mixture models to annotate image data, culminating in the *latent Dirichlet allocation* model (LDA). It specifies the following joint distribution of regions, words and latent variables (θ, z, y) :

$$p(r, w, \theta, z, y) = p(\theta | \alpha) \left(\prod_{n=1}^N p(z_n | \theta) p(r_n | z_n, \mu, \sigma) \right) \cdot \left(\prod_{m=1}^M p(y_m | \theta) p(w_m | y_m, \beta) \right). \quad (3.7)$$

This model assumes that a Dirichlet distribution θ (with α as its parameter) generates a mixture of latent factors, z and y . Image regions r_n are modelled with Gaussians with mean μ and covariance σ , where words w_n follow a multinomial distribution with a β parameter. This mixture of latent factors is then used to generate words (y variable) and regions (z variable). The EM algorithm estimates this model, and the inference of $p(w | r)$ is carried out by variational inference. The LDA model provides a clean probabilistic model for annotating images with multiple keywords. It combines the advantages of probabilistic clustering for dimensionality reduction with an explicit model of the conditional distribution from which image keywords are generated. Barnard et al. (2003), improved and compared the three previously discussed models: the *machine translation* model by Duygulu et al. (2002), the *hierarchical aspect model* by Barnard and Forsyth (2001) and the LDA model by Blei and Jordan (2003).

Also in this family of approaches, Quattoni, Collins and Darrel (2007) employ image captions to generate a new intermediate representation of images. The problem is split into many auxiliary problems and a core problem. For the auxiliary problems a structure is learned that jointly models captions and images. A new representation is obtained by factorizing parameters of the auxiliary models (SVD decomposition) and the core problem learns the keyword models in the resulting space.

Hierarchical and Network Models

The above approaches assumed a minimal relation between the various elements of an image (blobs or tiles). In semantic-multimedia analysis concepts are inter-dependent, for example if a house is detected in a scene, then the probability of existing windows and doors in the scene are boosted, and vice-versa. In other words, when inferring the probability of a set of inter-dependent random variables, their probabilities are iteratively modified until an optimal point is reached. To avoid instability, loops must exist over a large set of random variables, see (Pearl 1988). Most of the

papers discussed below model keywords and data (words and blobs or tiles) as a set of inter-dependent random variables connected in a hierarchical or network model.

Different graphical models have been implemented in computer vision to model the appearance, spatial relations and co-occurrence of local parts. Li and Wang (2003) characterise the images with a hierarchical approach at multiple tiling granularities (i.e., each tile in each hierarchical level is subdivided into smaller sub-tiles). A colour and texture feature vector represents each tile. The texture features represent the energy in high-frequency bands of wavelet transforms. They represent each keyword separately with two-dimensional multi-resolution hidden Markov models. This method achieves a certain degree of scale invariance due to the hierarchical tiling process and the two-dimensional multi-resolution hidden Markov model.

Markov random fields and hidden Markov models are the most common generative models that learn the joint probability of the observed data (\mathbf{X}) and the corresponding labels (\mathbf{Y}). These models divide the image into tiles or regions (other approaches use contour directions but these are outside the scope of our discussion). A probabilistic network then models this low-level division, where each node corresponds to one of these tiles or regions and its label. The relation between nodes depends on the selected neighbouring method. The model has the following mathematical expression:

$$p(x, w) = \frac{1}{Z} \prod_i \left(\phi_i(x_i, w_i) \prod_{j \in N_i} \psi_{i,j}(w_i, w_j) \right), \quad (3.8)$$

where i indexes the image tiles, j indexes the neighbours of the current i tile, ϕ_i is the potential function of the current tile x_i and its possible labels w_i , and $\psi_{i,j}$ is the interaction function between the current tile label and its neighbours. Figure 3.4 illustrates the Markov random field framework.

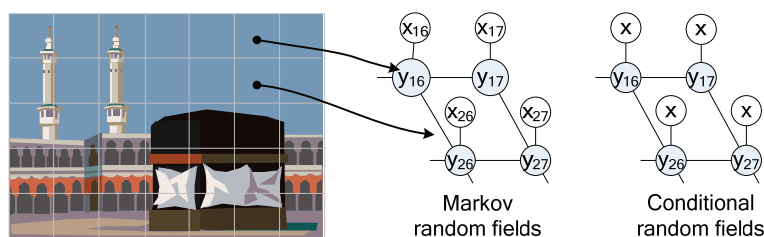


Figure 3.4. Two different types of random fields.

The Markov condition implies that a given node only depends on its neighbouring nodes. This condition constitutes a drawback for these models because only local relationships are incorporated into the model. This makes it highly unsuitable for capturing long-range relations or global characteristics. To circumvent this limitation Kumar and Herbert (2003b) propose a multi-scale

random field (MSRF) as a prior model on the class labels on the image tiles. This model implements a probabilistic network that can be approximated by a 2D hierarchical structure such as a 2D-tree. A multi-scale feature vector captures the local dependencies in the data. The distribution of the multi-scale feature vectors is modelled as a mixture of Gaussians. The features were specifically selected to detect human-made structures, which are the only types of objects that are detected.

Kumar and Herbert's (2003a) second approach to this problem is based on discriminative random fields, an approach inspired on conditional random fields (CRF). CRFs defined by Lafferty, McCallum, and Pereira (2001) are graphical models initially proposed for text information extraction, which are applied to visual information analysis in this approach. More generally, a CRF is a sequence-modelling framework based on the conditional probability of the entire sequence of labels (\mathbf{Y}) given the all image (\mathbf{X}). CRFs have the following mathematical form

$$p(w | x) = \frac{1}{Z} \prod_i \left(\phi_i(w_i, x) \prod_{j \in N_i} \psi_{i,j}(w_i, w_j; x) \right), \quad (3.9)$$

where i indexes the image tiles, j indexes the neighbours of the current i tile, ϕ_i is the association potential between the current tile and the image label, and $\psi_{i,j}$ is the interaction potential between the current tile and its neighbours (note that it is also dependent on the image label). The authors showed that this last approach outperformed their initial proposal of a multi-scale random field as well as the more traditional MRF solution in the task of detecting human-made structures.

He et al. (2004) combine the use of a conditional random field and data at multiple scales. Their multi-scale conditional random field (mCRF) is a product of individual models, each model providing labelling information from different aspects of the image: a classifier that looks at local image statistics; regional label features that look at local label patterns; and global label features that look at large, coarse label patterns. The mCRF is shown to detect several types of concepts (sky, water, snow, vegetation, ground, hippopotamus and bear) with classification rates better than a traditional Markov random field.

Murphy, Torralba and Freeman (2003) suggest solving the problem of information extraction (as an object recognition problem) by recurring to the scene context (image as a whole) as extra information. They aim to extract the type of scene and its objects including their position with wavelet based features and individual one-versus-all object classifiers and scene type discriminators. The individual object classifiers output is combined in a CRF for jointly solving the task of object recognition and scene detection. Their model explicitly assumes the inter-dependence between objects and scenes by assuming that object presence is conditionally independent given the scene.

Torralla, Murphy, and Freeman (2004) present an improvement over this approach with boosting to learn the conditional random field.

Quattoni, Collins, and Darrell (2004) extend the CRF framework to incorporate hidden variables and combine a class conditional CRFs into a unified framework for part-based object recognition. The features are extracted from special regions that are obtained with the scale-invariant feature transform (SIFT), see (Lowe 1999). The SIFT detector finds points in locations at scales where there is a significant amount of variation. Once a point of interest is found, the region around it is extracted at the appropriate scale. The features from this region are then computed and plugged into the CRF framework. The advantage of this method is that it needs a smaller number of tiles/regions by eliminating redundant tiles/regions and selecting special regions where high-frequency energy is high.

One should note that all these approaches require a ground-truth at the level of the image tiles/regions as is common in computer vision. This is not what is traditionally found in multimedia information retrieval datasets, where the ground-truth exists rather at a global level.

Knowledge based Models

The previous methods only have visual features as training data to create the statistical models in the form of a probabilistic network. This training data is – most of the times – limited, and the model accuracy can be improved by other sources of knowledge. Prior knowledge can be added to a model either by a “human expert”, who states the relations between concept variables (nodes in a probabilistic network), or by an external knowledge base to infer the concepts relations, e.g., a linguistic database, WordNet, see Figure 3.5.

Tansley (2000) introduces a multimedia thesaurus in which media content is associated with appropriate concepts in a semantic layer composed of a network of concepts and their relations. The process of building the semantic layer uses Latent Semantic Indexing to connect images to their corresponding concepts, and a measure of each correspondence (image-concept) is taken from this process. After that, unlabelled images (test images) are annotated by comparing them with the training images using a k -nearest-neighbour classifier. Since the concepts’ inter-dependences are represented in the semantic layer, the probability concepts computed by the classifier is modified by the others concepts.

Other authors have explored not only the statistical inter-dependence of context and objects but also other knowledge, not present in multimedia data, that humans use to understand (or predict) new data. Srikanth et al. (2005) incorporated linguistic knowledge from WordNet, see (Miller 1995), to deduce a hierarchy of terms from the annotations. They generate a visual vocabulary based on the semantics of the annotation words and their hierarchical organization in the WordNet ontology.

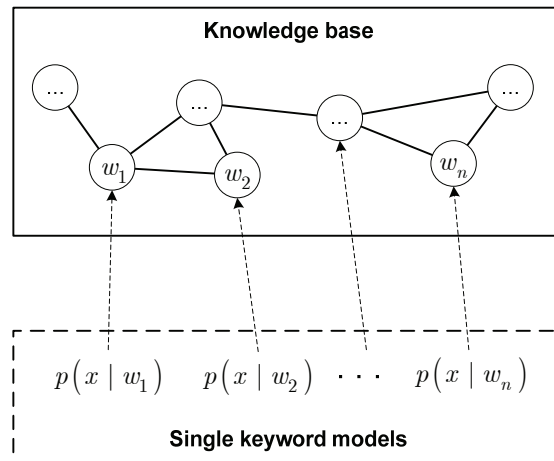


Figure 3.5. Knowledge based models.

Benitez and Chang (2002) and Benitez (2005) took this idea further and suggested a media-ontology (MediaNet) to help to discover, summarise and measure knowledge from annotated images in the form of image clusters, word senses and relationships among them. MediaNet, a Bayesian-network-based multimedia knowledge representation framework, is composed by a network of concepts, their relations and media exemplifying concepts and relationships. MediaNet integrates classifiers to discover statistical relationships between concepts. WordNet (Miller 1995), is used to process image annotations by stripping out unnecessary information. The summarization process implements a series of strategies to improve image description quality, e.g., using WordNet and image clusters to disambiguate annotation terms (images in the same clusters tend to have similar textual descriptions). Benitez (2005) also proposes a set of measures to evaluate the knowledge consistency, completeness and conciseness.

3.2.2 Text Analysis

Text categorization models pre-process data by removing stop-words and rare words, stemming, and finally term-weighting. Due to the high-dimensional feature space of text data most text categorization algorithms are linear models such as naïve Bayes (McCallum and Nigam 1998), maximum entropy (Nigam, Lafferty and McCallum 1999), Support Vector Machines (Joachims 1998), regularized linear models (Zhang and Oles 2001), and Linear Least Squares Fit (Yang and Chute 1994). Joachims (1998) applies SVMs directly to the text terms. Text is ideal for applying SVMs without the need of a kernel function because data is already sparse and high-dimensional. Linear models fitted by least squares such as the one by Yang and Chute (1994) offer good precision, and in particular regularized linear methods, such as the one proposed by Zhang and Oles (2001), perform similarly to SVMs, with the advantage of yielding a probability density model. The maximum entropy classification model proposed by Nigam, Lafferty and McCallum (1999)

defines a set of features that is dependent on the class being evaluated.

Yang (1999), and Yang and Liu (1999) have compared a number of text classification algorithms and reported their performances on different text collections. Their results indicate that k -nearest-neighbour, SVMs, and LLSF are the best classifiers. Note that nearest neighbour approaches have certain characteristics, see (Hastie, Tibshirani and Friedman 2001), that make them computationally expensive to handle large-scale indexing tasks.

3.2.3 Audio Analysis

Audio analysis becomes a very important part of the multi-modal analysis task when processing TV news, movies, sport videos, etc. Different types of sounds can populate the sound track of a multimedia document, with the most common types being speech, music and silence.

In most TV programs and sport videos these events do not overlap, but in narratives (movies and soap operas) these events frequently occur simultaneously. Akutsu, Hamada, and Tonomura (1998) present an audio-based approach to video indexing by detecting speech and music independently (even in the case where they occur simultaneously). Their framework is based on a set of heuristics over feature histograms and corresponding thresholds. With a similar goal, Naphade and Huang (2000) define a generic statistical framework based on hidden Markov models, see (Rabiner 1989), to classify audio segments into *speech*, *silence*, *music* and *miscellaneous* and their co-occurrences. By creating an HMM for each class and every combination of classes the authors achieved a generic framework capable of modelling different audio events with high accuracy.

Lu, Zhang and Jiang (2002) propose methods to segment audio and classify each segment as *speech*, *music*, *silence* and *environment sound*. A k -nearest neighbour model is used at the frame level followed by vector quantization to discriminate between speech and non-speech. A set of threshold-based rules is used to discriminate between silence, music and environment sound. The authors also describe a speaker change detection algorithm based on Gaussian-mixtures models (GMM); this algorithm continuously compares the model of the current speaker's speech with a model dynamically created from the current audio frame. After a speaker change has been detected, the new GMM replaces the current speaker's GMM.

Another important audio analysis task is the classification of the musical genre of a particular audio segment. This can capture the type of emotion that a movie director wants to communicate, e.g., stress, anxiety or happiness. Tzanetakis and Cook (2002) describe their work on categorizing music as *rock*, *dance*, *pop*, *metal*, *classical*, *blues*, *country*, *hip-hop*, *reggae* or *jaz̄z* (jazz and classical music had more sub-categories). In addition to the traditional audio features, they also use special features to capture rhythmic characteristics and apply simple statistical models such as GMM and k -NN to model each class's feature histogram. Interestingly, the best reported classification precision (61%)

is on the same range as human performance for genre classification (70%).

All these approaches work as a single class model of individual classes/keywords. Note that the hidden Markov model is in fact a probabilistic network for modelling a single temporal event that corresponds to a given concept/keyword. So, even though it is a network model, it is used as a single class model.

3.3 Multi-Modal Analysis

Multimedia content can be indexed in many ways and each index can refer to different modalities and/or different parts of the multimedia piece. In the previous algorithms, audio and visual modalities were processed independently to detect semantic entities. These semantic entities are represented in different modalities capturing different aspects of that same reality. Multimedia content is composed by the visual track, sound track, speech track and text. All these modalities are structured in the best way to communicate information.

Structure analysis must be executed before extracting semantic information from a multimedia document. Given a multimedia document, an analysis algorithm must first compute the single-media segments and their relations. Later, analysis algorithms process these segments to extract semantic information. The following sections will discuss some of the approaches used in these steps.

3.3.1 Structure Analysis

In Chapter 1 we defined multimedia and provided two examples of such type of content: HTML and video content. Both types of documents structure information across distinct modalities in different ways: while in HTML information is presented in a spatially layout, in video the information is presented in a temporal line. Thus, different structure analysis algorithms are required to split the multi-modal document into its meaningful segments of single-media information.

Segmenting multimedia documents is outside the scope of this thesis, and we only address video content.

Spatial Structure Analysis

Due to the advent of the WWW, the HTML format is a common electronic resource available nowadays, making it a natural target for information extraction systems. An HTML document is often rich in media objects and layout information. Compared to plain text, HTML adds a layer of metadata that mostly contains spatial (layout) information, on top of the text. This type of content requires a spatial structure (layout) analysis to parse the document specific format and identify its

blocks of information. Figure 3.6, taken from (Cai et al. 2003), illustrates a layout analysis that generates visual segments of an HTML page.

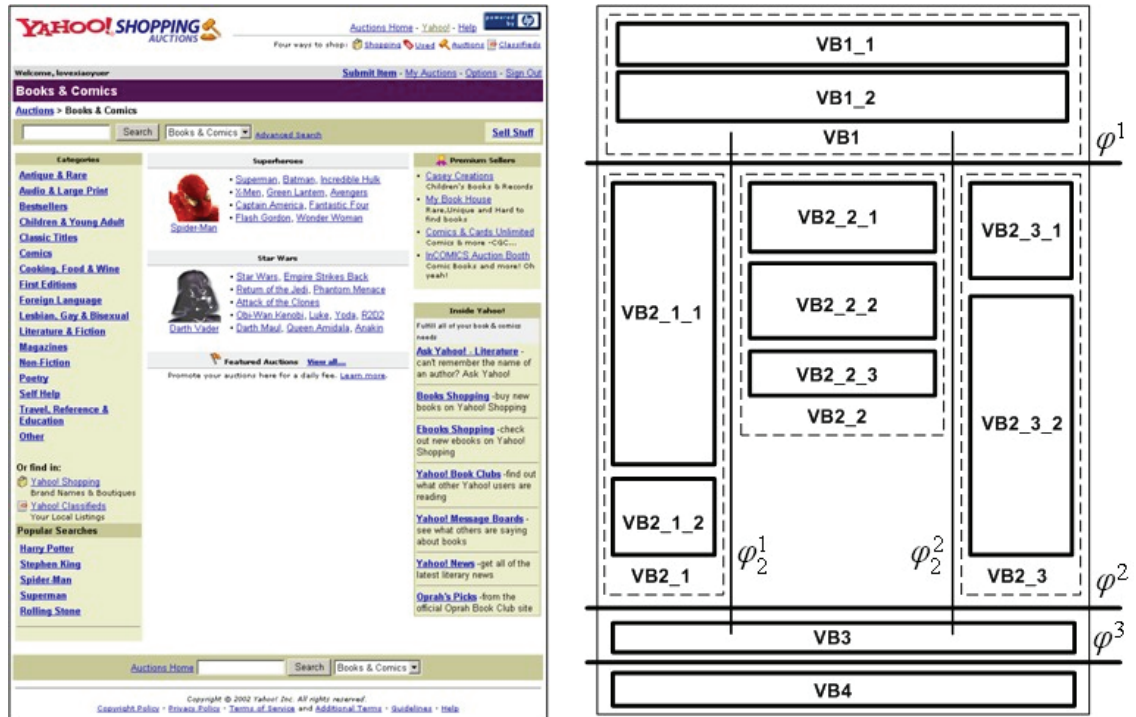


Figure 3.6. Spatial structure of an HTML document, (Cai et al. 2003).

Temporal Structure Analysis

Video content is composed by temporal scenes concerning meaningful information. From these scenes, example key-frames and automatic speech recognition can be applied to extract text and image content. Thus, synchronization and the strategy to combine the multi-modal patterns is the key issue on multi-modal analysis. The approaches described in this section explore the multi-modality statistics of semantic entities, e.g., pattern synchronization.

Video documents are temporally structured at two levels of abstraction: syntactic and semantic levels, see Figure 3.7. At the syntactic level the video is segmented into shots (visual or audio shots) that form a uniform segment, e.g., visually similar frames; representative key-frames are extracted from each shot, and scenes group neighbouring similar shots into a single segment. The segmentation of video into its syntactic structure has been widely studied, e.g., (Brunelli et al., 1999) and (Wang et al., 2000).

At the semantic level, annotations of the key-frames and shots with a set of labels indicate the presence of semantic entities, their relations and attributes, (agent, object, event, concept, state, place and time, see (Benitez et al., 2002) for details). Further analysis allows the discovery of stories, sub-stories, or genres.

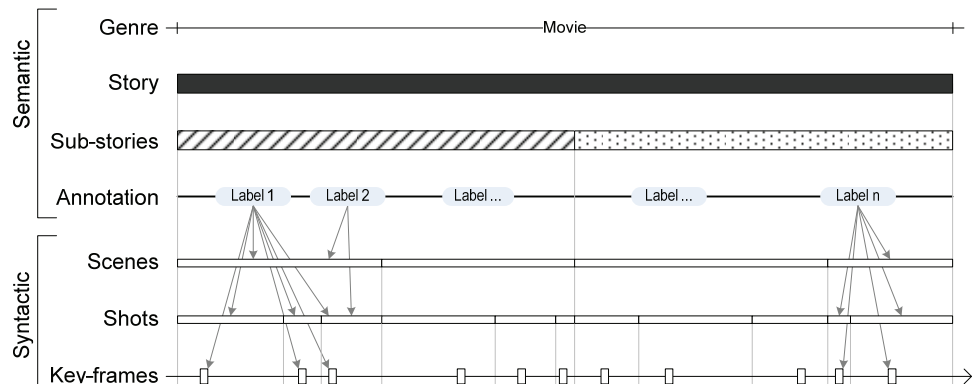


Figure 3.7. Temporal structure of a video document.

Shot and scene semantic analysis introduce the time dimension, which adds temporal frames resulting in more information to help the analysis. In order to take advantage of the sequential nature of the data the natural choices of algorithms are based on hierarchical models or network models. However, as we shall see, some simple approaches can also offer competitive results.

The scope of this section is the family of semantic-multimedia analysis algorithms that automate the multimedia semantic annotation process. The analysis at the shot and scene level independently considers the audio and visual modalities, and then multi-modal semantic analysis. In the following sections we will review papers on semantic-video analysis. A thorough review of multimedia semantic indexing has been published by (Snoek and Worring 2005b).

3.3.2 Heuristic based Models

Many of the visual video analysis methods are based on heuristics that are deduced empirically. However, statistical methods are more common when considering more than one modality. Most of the following papers explore the temporal evolution of features to semantically analyse video content, e.g., shot classification, logical units, etc.

A first approach to this task is to detect events by monitoring histograms and trigger a detector if a given threshold is exceeded. These methods are particularly adequate for sport videos because broadcast TVs follow a set of video production rules, which result in well defined semantic structures that ease the analysis of the sport videos.

Basketball video analysis was addressed by Tan et al. (2000) who introduced a model for estimating the camera movements (pan, tilt and zoom) from the motion vectors of compressed video. The authors further show how camera motion histograms can be used to discriminate different basketball shots. Prior to this, the video is segmented into shots based on the evolution of the intensity histogram across different frames. Shots are detected if the histogram exceeds a predefined threshold. They are then discriminated based on (a) the accumulated histogram of

camera motion direction (*fast breaks* and *full court advances*), (b) the slope of this histogram (*fast breaks* or *full court advances*), (c) sequence of camera movements (*shots at the basket*) and (d) persistence of camera motion (*close-ups*).

Other heuristic methods deploy colour histograms, shot duration and shot sequences to detect events and classify shots of different sports such as football (Ekin, Tekalp and Mehrotra 2003), American football (Li and Sezan 2003), or tennis (Luo and Hwang 2003).

3.3.3 Statistics based Models

Depending on the level of analysis depth, some approaches keep time dependence modes of an entire shot time, others just average the output of single key-frames analysis; while others exploit the time dimension to improve low-level feature extraction, e.g., shapes obtained by segmentation are more accurate. As we shall see now, most of the approaches described in Section 3.2.1 can also be applied to the visual analysis of video content.

Single Class Models

In TV news videos, text is the fundamental modality with the most important information. Westerveld et al. (2003) build on their previous work, see (Westerveld and de Vries 2003b) described above, to analyse the visual part and add text provided by an Automatic Speech Recognition (ASR) system. The authors further propose a visual dynamic model to capture the visual temporal characteristics. This model is based on the Gaussian mixture model estimated from the DCT blocks of the frames around each key-frame in the range of 0.5 seconds. This way the most significant moving regions are represented by this model with an evident applicability to object tracking. The text retrieval model evaluates a given $Shot_i$ for the queried keywords $Q = \{q_1, q_2, q_3, \dots\}$ by:

$$\begin{aligned} \text{RSV}(Shot_i) = \frac{1}{|Q|} \sum_{k=1}^{|Q|} \log [& \lambda_{Shot} p(q_k | Shot_i) + \\ & + \lambda_{Scene} p(q_k | Scene_i) + \lambda_{Coll} p(q_k)]. \end{aligned} \quad (3.10)$$

This measure evaluates the probability that one or more queried keywords appear in the evaluated shot in the scene prior. The λ variables correspond to the probabilities of respective weights. This function, inspired by language models, creates the scene-shot structure of video content. The visual model and the text model are combined under the assumption that they are independent, thus their probabilities are simply multiplied. The results with both modalities are reported to be better than using just one.

Support vector machines have been widely used in retrieval scenarios. Zheng et al. (2008) brought a fresh look approach to multimedia concept detection by explicitly tackling it as a ranking problem, which is addressed by a generative approach (Relevance Vector Machine) and compared to a discriminative approach (SVM). It tackles a valid, and many times forgotten, point regarding the fact that using SVMs for ranking is not a mathematically sound solution.

Translation Models

Latent semantic analysis was applied to video retrieval by Souvannavong et al. (2003). Recall that LSA algorithm builds a matrix M of word occurrences in documents, and then the SVD of this matrix is computed to obtain a canonical space. The problem with multimedia content is that there is often no text corpus and hence no vocabulary. A vector quantization technique (k -means) returns a codebook of blobs, the vocabulary of blobs from the key-frames. In the singular feature space a k -NN ($k = 20$) and a Gaussian mixture model technique are used to classify new videos. The comparison of the two techniques shows that GMMs perform better when there is enough data to correctly estimate the 10 components. The k -NN algorithm has the disadvantages of every nonparametric method, where the model is the training data, and therefore training can take considerable time.

Bag-of-features approaches have many factors that affect their performance: choice of features, vocabulary size, kernels, and weighting scheme. Jiang, Ngo and Yang (2007) deployed a bag-of-features system that assessed the optimal combination of these factors. Experiments on a TRECVID collection proved it to be very successful at the cost of a large number of tuning parameters.

In (Li et al. 2003) two media types are considered (visual and audio). For every individual modality a set of appropriate features is initially determined and their variation in time is recorded. Then, based on appropriate statistical methodologies, transformation matrices between the audio and the visual feature spaces are estimated. In (Wu et al. 2004) the main objective is to statistically examine the n individual modalities that are concerned, and to compute a set of D , where $D < n$, statistically independent (or almost independent) mixed modalities. For that purpose, statistical algorithms (like PCA, ICA) are implemented in order both to reduce the feature space and to compute the D independent modalities.

Hierarchical and Network Models

Explicitly modelling of synchronization and time relations between different patterns are the base of Snoek and Worring's (2005a) approach. They propose a multimedia analysis framework based on Allen's (1983) temporal interval relations. Allen showed that to maintain temporal knowledge about any two events only a small set of relations are needed to represent their temporal

relations. These relations, now applied to audio and visual patterns, are: *precedes*, *meets*, *overlaps*, *starts*, *during*, *finishes*, *equals* and *no-relation*. The multimedia analysis framework can include context and synchronization of heterogeneous information sources involved in multimodal analysis. Initially, the optimal pattern configuration of temporal relations of a given event is learnt from training data by a standard statistical method (maximum entropy, decision trees and SVM). New data are classified with the learned model. The authors evaluate the event detection on soccer video (*goal*, *penalty*, *yellow card*, *red card* and *substitution*) and TV news (*reporting anchor*, *monologue*, *split-view* and *weather report*). The difference between the different classifiers (maximum entropy, decision trees and SVM) appears to be not significant.

Sport video analysis can be greatly improved with multi-modal features, for example the level of excitement expressed by the crowd noise can be a strong indicator of certain events (fault, goal, goal miss, etc). Leonardi, Migliotari and Prandini (2004) take this into account when designing a multi-modal algorithm to detect goals in football videos. A set of visual features from each shot is fed to a controlled Markov chain to evaluate their temporal evolution from one shot to the next one. The Markov chain has two states corresponding to the goal state and to the non-goal state. The visual analysis returns the positive pair-shots and the shot audio loudness is the criterion to rank the pair-shots. Thus, the two modalities are used sequentially. Results show that audio and visual modalities together improve the average precision when compared to only the audio case, see (Leonardi, Migliotari and Prandini 2004).

Luo and Hwang's (2003) statistical framework tracks objects within a given shot with a dynamic Bayesian network and classifies that shot from a coarse-grain to a fine-grain level. At the coarse-grain level a key-frame is extracted from a shot every 0.5 seconds. From these key-frames motion and global features are extracted and their temporal evolution is modelled with a hierarchical hidden Markov model (HHMM). Individual HHMMs (a single class model approach) capture a given semantic shot-category. At the fine-grain level analysis Luo and Hwang (2003) work heavily with object recognition and tracking. After the coarse-grain level analysis segmentation is performed on the shots to extract visual objects. Then invariant points are detected in each shape to track the object movement. These points are fed to a dynamic Bayesian network to model detailed events occurring within the shot, e.g., human body movements in a golf game.

Knowledge based Models

Naphade and Huang (2001) characterise single-modal concepts (e.g., indoor/outdoor, forest, sky, water) and multi-modal concepts (e.g., explosions, rocket launches) with Bayesian networks. The visual part is segmented into shots, see (Naphade et al. 1998), and from each key-frame a set of low-level features is extracted (colour, texture, blobs and motion). These features are then used to estimate a Gaussian mixture model of multimedia concepts at region level and then at frame level.

The audio part is analysed with the authors' algorithm described above, see (Naphade and Huang 2000). The outputs of these classifiers are then combined in a Bayesian network to improve concept detection. Their experiments show that the Bayesian network improves the detection performance over individual classifiers.

IBM research by Adams et al. (2003) extend the approach of Naphade and Huang (2001) by including text from Automatic Speech Recognition as a third modality and by using Support Vector Machines to combine the classifier outputs. The comparison of these two combination strategies showed that SVM (audio, visual and text) and Bayesian networks (audio and visual) perform equally well. However, since in the later case, speech information was ignored one might expect that Bayesian networks can in fact perform better. More details about this work can be found in (Naphade and Smith 2003), (Natsev, Naphade and Smith 2003) and (Tseng et al. 2003).

3.4 Discussion

Single-Media Analysis

The described algorithms vary in many different aspects, such as in their low-level features, segmentation methods, feature representation, modelling complexity or required data. While some concepts require large amounts of training data to estimate its model, e.g., car, others are very simple and require just a few examples, e.g., sky. So, we advocate that different approaches should be used for different concept complexities.

Single class models assume that concepts are independent and that each concept has its own model. These are the simplest models and the ones with better accuracy.

Translation models, hierarchical models, and network models capture a certain degree of the concept inter-dependence (co-occurrence) from the information present in the training data. The difference between the models is linked to the degree of inter-dependence that can be represented by the model. In practice, when inter-dependency information is incorporated in the model it also inserts noise in the form of false inter-dependency, which can cause a decrease in performance. So, the theoretical advantage of these models is in practice reduced by this effect.

All these models rely exclusively on visual low-level features to capture complex human concepts and correctly predict new unlabelled data. The training data is – most of the times – limited, and the model accuracy can be improved by other sources of knowledge. Srikanth et al. (2005) and Benitez (2005) are two of the few proposals that exploit prior knowledge external to the training data to capture the inter-dependent (co-occurrence) nature of concepts.

At this time, knowledge based models seem the most promising semantic analysis algorithms for information retrieval. Text information retrieval has already shown a great improvement over

exclusively statistical models when an external linguistic database was used, see (Harabagiu et al. 2000). I predict that multimedia retrieval will go through a similar progress but at a slower pace because there is no “multimedia ontology” offering the same knowledge base as WordNet offers to linguistic text processing.

Multi-Modal Analysis

When considering multi-modal content a new and very important dimension is added: time. Time adds a lot of redundancy that can be effectively explored to achieve a better segmentation and semantic analysis. The most interesting approaches consider time either implicitly, e.g., (Westerveld et al. 2003), or explicitly, e.g., (Snoek and Worring 2005a).

Few papers show a deeper level of multi-modal combination than Snoek and Worring (2005a) and Naphade and Huang (2001). The first explicitly explores the multi-modal co-occurrence of patterns resulting from the same event with temporal relations. The later integrates multi-modal patterns in a Bayesian network to explore patterns co-occurrences and concept inter-dependence.

Natural language processing experts have not yet applied all the techniques from text to the video’s extracted speech. Most approaches to extract information from text and to combine it with audio and visual extracted information are very simple, such as a simple product between the probabilities of different modalities classifiers.

3.5 Conclusions

Semantic-multimedia analysis for retrieval applications has delivered its first promises, and many novel contributions will be done over the next years. To achieve a more comprehensive understanding of the field, we conducted a thorough research of previous works and organized them according to its media types and into different families of algorithms:

- **Single class models**
- **Translation models**
- **Hierarchical and network models**
- **Knowledge based models**

Major developments in semantic-multimedia analysis algorithms will probably be related to knowledge based models and multi-modal fusion algorithms. Future applications might boost knowledge based models research by enforcing a limited application domain (i.e. a constrained

knowledge base). Examples of such applications are football games summaries, and mobile photo albums.

Multi-modal analysis algorithms have already proven to be crucial in semantic-multimedia analysis. Large developments are expected in multi-modal analysis owing to its relative novelty where several problems wait to be explored and owing to the TRECVID conference series that pushes forward this research area through a standard evaluation methodology and a rich multimedia collection. The limited research in multi-modal fusion algorithms is due to the different expert knowledge that is required to deal with the different modalities (image analysis, audio and speech analysis, natural language processing).

A Multi-Modal Feature Space

4.1 Introduction

Demand for techniques that handle both text and image based documents is increasing with the wide spread of search applications. It is impossible to conceive nowadays a world without systems that allow us to search for specific news articles, scientific papers, or information in general. Users want more: they want to have the same retrieval model that would allow to search for text documents, visual documents, or documents with both media, e.g., photographs with captions, video shots (key-frames and speech). To achieve this, a new breed of information retrieval models is required: one that seamlessly integrates heterogeneous data. Thus, in this thesis we assume that in any given collection \mathcal{D} of N multimedia documents

$$\mathcal{D} = \{d^1, d^2, \dots, d^N\}, \quad (4.1)$$

each document is characterized by a vector

$$d^j = (d_T^j, d_V^j, d_W^j), \quad (4.2)$$

composed by a feature vector d_T describing the text part of the document, a feature vector d_V describing the visual part of the document, and a keyword vector d_W describing the semantics of the document. More specifically we have:

- The feature vector d_T contains text based features such as text terms obtained via a stemmer, bag-of-word, part-of-speech or named entities
- The feature vector d_V contains low-level visual features such as texture, colour or shape

- The feature vector d_W contains keyword confidence scores concerning the detection of the corresponding concept

Algorithms and techniques to compute low-level text and visual features are widely studied, and several algorithms exist to extract them. Keyword features representing multimedia information have a less consensual solution because of the ambiguity and subjectivity of the information that they try to describe – the semantic content of a multimedia document. The semantic description of multimedia information, the feature vector d_W , is the core topic of this thesis. To describe the semantics of multimedia information we define the set

$$\mathcal{W} = \{w_1, \dots, w_L\} \quad (4.3)$$

as a vocabulary of L keywords. Keywords are linguistic representations of abstract or concrete concepts that we want to detect in multimedia documents. The feature vector d_W is formally defined as

$$d_W^j = \left(d_{W,1}^j, d_{W,2}^j, \dots, d_{W,L}^j \right) \quad (4.4)$$

where each component $d_{W,t}^j$ is a score indicating the confidence that keyword w_t is present in that particular document. The concepts may not be explicitly present in multimedia information, methods are required to compute the likelihood that the keyword is actually present in the multimedia document.

Equation (4.2) shows us the other information that we have about documents: text and visual feature. Thus, to compute the components of the keyword vector d_W^j we shall use text and visual feature data. This leads us to the definition of each component of the keyword vector as

$$d_{W,t}^j = p\left(y_t^j = 1 \mid d_T^j, d_V^j\right), \quad (4.5)$$

where the random variable $y_t^j = \{1, 0\}$ indicates the presence/not-presence of keyword w_t on document d^j given its text feature vector d_T^j and visual feature vector d_V^j . This enables the semantic indexing of multimedia content which allows users to *submit the same query to search for text documents, visual documents, or documents with both media, e.g., photographs with captions, video shots (key-frames and speech)*. Equation (4.2) integrates heterogeneous representations of a multimedia document (text, image and semantic) and Equation (4.5) will make multimedia information searchable with the same type of queries for all type of media.

In Chapters 4 and 5 we shall address the problem of estimating a statistical model for Equation (4.5). I shall propose a statistical framework that can simultaneously model text-only documents,

image-only documents, and documents with both text and images. I will follow a statistical learning theory approach to solve the semantic multimedia analysis problem, more specifically as a multi-label classification problem. Figure 4.1 illustrates the traditional framework: statistical models are learned from the training data of keyword, and new data are labelled with the trained models. For excellent references on statistical learning theory and pattern recognition see (Hastie, Tibshirani and Friedman 2001) and (Duda, Hart and Stork 2001).

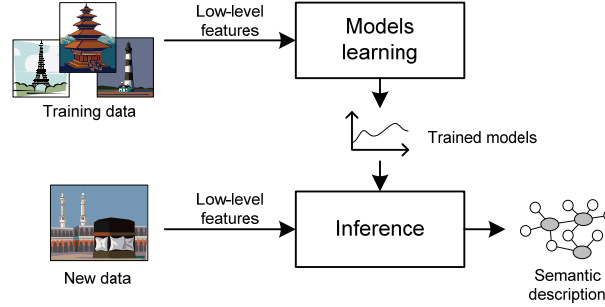


Figure 4.1. The traditional statistical learning theory framework.

4.2 Multi-Modal Keyword Models

Our first objective is to compute the components of keyword feature vectors d_W representing the semantics of multimedia documents. For this we will estimate and select a model $\beta_t \in \Theta$, from a set Θ of candidate models, that best represents the keyword w_t in terms of text data and visual data. We omit model β_t of keyword w_t from Equation (4.5) for notational simplicity. The expression can now be written as:

$$d_{W,t}^j = p(y_t^j = 1 | d_T^j, d_V^j, \beta_t) \quad (4.6)$$

The statistical model $\beta_t \in \Theta$ can assume many forms (e.g., nearest neighbour, neural networks, linear models, support vector machines) according to the family of algorithms and to the complexity of the specific algorithm within a particular family of algorithms. The choice of the family of algorithms is done by examining the requirements that multimedia information retrieval applications face in a real world scenario:

- Arbitrary addition and removal of keywords
- Easy update of existing keyword models with new training data
- Seamless integration of heterogeneous types of data

- Computationally efficient indexing of multimedia information
- Good retrieval effectiveness

The first two requirements concern an important practical aspect in large-scale multimedia indexes – the integrity of the index when keyword models are modified. When a keyword model is modified (added, removed or updated) the index can be affected in two ways: if keyword models are dependent then the entire index becomes obsolete; if keyword models are independent then only the part of the index concerning that keyword becomes obsolete. This leads to a solution where keyword models are independent so that a modification in one keyword model will have a minor influence on the indexes. Thus, presence of keywords shall be represented by Bernoulli random variables

$$p(y_t | \rho_t) = \rho_t^{y_t} (1 - \rho_t)^{1-y_t} \quad y_t = \{0,1\}, \quad (4.7)$$

where ρ_t is the probability of keyword w_t .

The remaining three requirements can be difficult to accommodate in a unique model: support of multi-modal information, be able to quickly index new multimedia content and to achieve a good accuracy. When modelling multi-modal keywords, one has to deal with both dense feature spaces and sparse feature spaces. On the one hand visual feature data can be very dense making its modelling difficult due to the irregular frontiers caused by concept cross-interference. Expanding the original feature space into higher-dimensional ones results in a sparser feature space where the modelling of the data can be made easier. On the other hand, text feature spaces are typically too sparse making its modelling difficult because there is not enough support data to estimate the details of concept models. In these situations we have to compress the feature space into a lower dimensional space where data is compressed into a more dense space. These transformations of the original feature space into a space where the data is optimally distributed is represented as

$$F(d_T^j, d_V^j) = (F_T(d_T^j), F_V(d_V^j)), \quad (4.8)$$

where $F_T(d_T^j)$ correspond to the text data transformation and $F_V(d_V^j)$ correspond to the visual data transformation. This renders the final expression for the components of keyword feature vectors as

$$d_{W,t}^j = p(y_t^j = 1 | F(d_T^j, d_V^j), \beta_t). \quad (4.9)$$

The transformation of multimedia document features only need to be computed once for all

keyword models, in other words, the transformation is independent of the keyword models. The interesting implication of this fact is that it can reduce the indexing computational complexity: because the transformation generates a high-dimensional space, one can limit the keyword model search space Θ to the family of linear models which have a very low computational complexity in the classification phase (but not necessarily in the learning phase). Besides the low computational complexity, linear models offer other interesting advantages: support of high-dimensional data (easy integration of heterogeneous data), naturally embedded background knowledge in the form of priors (ideal for keyword model update) and good accuracy (retrieval effectiveness).

In the remainder of Chapter 4 I will present and propose $F_T(d_T)$ and $F_V(d_V)$ the transformations of visual and text data. Chapter 5 presents linear models β_i to represent keywords.

4.3 Optimal Data Representation

The transformations $F_T(d_T)$ and $F_V(d_V)$ change the representation of the original text and visual feature spaces. As mentioned, transformations $F_T(d_T)$ and $F_V(d_V)$ will adopt specific strategies adequate to the characteristics of each type of data. However, in both cases there is the problem of selecting the optimal transformation from the large number of possible transformations and their varying complexities. In practice, the selection of the optimal transformation is equivalent to old questions like “*how many text features?*” and “*how many visual clusters?*” that are usually addressed by some heuristic method. In this section I shall formally address this problem.

The proposed feature space transformations are inspired by information theory: the space transformation F can be seen as a codebook composed by a set of $M = M_T + M_V$ codewords representing the data space. Given the codebook of a feature space one is able to represent all samples of that feature space as a linear combination of keywords from that codebook. Information theory (Cover and Thomas 1991) provides us with a set of information measures that not only assess the amount of information that one single source of data contains, but also the amount of information that two (or more) sources of data have in common. Thus, we employ the minimum description length criterion (Rissanen 1978), to infer the optimal complexity M_T and M_V of each feature space transformation $F_T(d_T)$ and $F_V(d_V)$. Note that I use the word “*optimal*” from an information theory point of view. The treatment of the model selection problem presented in this section is based on (Hastie, Tibshirani and Friedman 2001) and (MacKay 2004).

4.3.1 Assessing the Data Representation Error

The process of changing the original feature-space representation into the new representation with a given candidate transformation \hat{F} has an associated error. If we represent \hat{F} as the

estimated transformation, and G as the lossless transformation that we are trying to estimate, we can compute the mean-squared deviation between the estimated model and the desired response as the error

$$\begin{aligned} \text{Err}_{\mathcal{D}}(d) &= \mathbb{E}\left[\left(G(d) - \hat{F}(d)\right)^2\right] \\ &= \sigma_e^2 + \left(\mathbb{E}[\hat{F}(d)] - G(d)\right)^2 + \mathbb{E}\left[\hat{F}(d) - \mathbb{E}[\hat{F}(d)]\right]^2. \end{aligned} \quad (4.10)$$

The first term is the variance of the modelled process and cannot be avoided. The second term measures the difference between the true mean of the process and the estimated mean. The third term is the variance of the estimated model around its mean. The above expression can be written as:

$$\text{Err}_{\mathcal{D}}(d) = \sigma_e^2 + \text{Bias}^2(\hat{F}(d)) + \text{Variance}(\hat{F}(d)) \quad (4.11)$$

The more complex we make the candidate transformation \hat{F} the lower the bias but higher the variance. Equation (4.11) expresses the transformation bias-variance tradeoff: simple transformations can only represent the training data's coarse details (high bias) causing a high prediction error (low variance) because the transformation ignores important aspects of the data structure; complex transformations can represent training data structures in great detail (lower bias) but the prediction error increases (in variance) because the transformation do not generalise to other data.

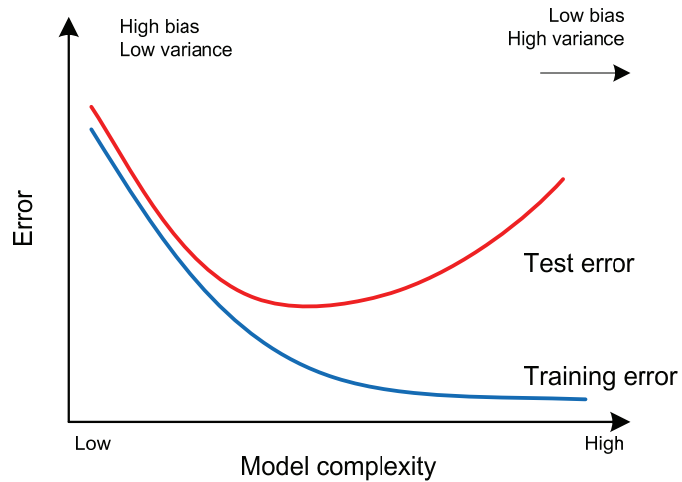


Figure 4.2. Bias-variance trade-off curve.

Figure 4.2 illustrates a typical bias-variance trade-off curve of the training sample error and the test sample error. The optimal transformation is the one that achieves the best generalization error

on the new unseen samples. There are two types of methods to select the transformation that has the best generalization error: empirical methods use validation data different from the training data to assess the model generalization error on the test data, e.g., cross-validation and bootstrap; criteria based methods provide an estimate of the model generalization error on the test data based on the error on the training data and the complexity the model, e.g., Bayesian Information Criterion. The minimum description length criterion is in the later group, and I chose it as the model selection criterion for feature space transformation.

4.3.2 The MDL Principle

Model selection is a widely studied subject, see (Hastie, Tibshirani and Friedman 2001), and the minimum description length (MDL) criterion is among the most common criteria of model selection. Rooted in information theory, the MDL principle was initially thought as a method to find the minimum number of bits required to transmit a particular message msg . To transmit this message a codebook cbk such as Huffman coding can be used to compress the message. Thus, the total number of bits required to transmit the message is

$$DL(msg, cbk) = DL(msg | cbk) + DL(cbk), \quad (4.12)$$

corresponding to the description length of the message msg encoded with the codebook cbk plus the description length of the codebook cbk . The MDL principle says that the optimal trade-off between these two quantities is achieved with the codebook cbk_{min} that minimizes the above expression. The minimum description length is written as

$$MDL(msg) = DL(msg | cbk_{min}) + DL(cbk_{min}), \quad (4.13)$$

where cbk_{min} is the optimal codebook that allows the message msg to be transmitted with the minimum number of bits.

The relation between the MDL criterion and the problem of model selection is straightforward: it assesses the trade-off between the data likelihood (the message) under a given model (the codebook) and the complexity of that model. In the problem we are addressing, the data \mathcal{D} will be transformed into a new feature-space by a transformation \hat{F} . Hence, Equation (4.12) is written as the sum of the likelihood of the data \mathcal{D} on the new feature space and the complexity of the feature-space transformation \hat{F} . Formally, we have

$$DL(\hat{F}_i, \mathcal{D}) = -\sum_{d \in \mathcal{D}} \log p(d | \hat{F}_i) + \frac{n_{pars}}{2} \cdot \log N, \quad (4.14)$$

where n_{pars} is the number of parameters of the transformation \hat{F} , and N is the number of samples in the training dataset. Hence, the MDL criterion is designed “to achieve the best compromise between likelihood and ... complexity relative to the sample size”, (Barron and Cover 1991). Finally, the optimal feature-space transformation is the one that minimizes Equation (4.14), which results in

$$F = \arg \min_{\hat{F}} DL(\hat{F}, \mathcal{D}). \quad (4.15)$$

The MDL criterion provides an estimate of the model error on the test data. Note that it is not an absolute estimate – it is only relative among candidate models. To evaluate the set Θ of candidate models and to better assess the characteristics of each model relatively to others we can compute the posterior probability of each model,

$$P(F_n | \mathcal{D}) = \frac{e^{-\frac{1}{2}DL(F_n)}}{\sum_{i=1}^{|\Theta|} e^{-\frac{1}{2}DL(F_i)}}. \quad (4.16)$$

The minimum description length approach is formally identical to the Bayesian Information Criterion but is motivated from a Bayesian perspective, see (MacKay 2004).

4.4 Dense Spaces Transformations

Some of the input feature spaces (depending on its media type) can be very dense making its modelling difficult due to cross-interference between classes. Expanding the original feature space into higher-dimensional ones results in a sparser feature space where the modelling of the data can be easier. This technique is applied by many related methods such as kernels. The discussion section of the next chapter will discuss these relationships.

The low-level visual features that I use are dense and low-dimensional: hence, keyword data may overlap thereby increasing the cross-interference. This means that not only the discrimination between keywords is difficult but also the estimation of a density model is less effective due to keyword data overlapping. One solution is to expand the original feature space into a higher-dimensional feature space where keywords data overlap is minimal. Thus, we define F_V as the transformation that increases the number of dimensions of a dense space with m dimensions into an optimal space with k_V dimensions

$$\mathbb{F}_V(d_{V,1}, \dots, d_{V,m}) = \begin{bmatrix} f_{V,1}(d_{V,1}, \dots, d_{V,m}) \\ \vdots \\ f_{V,k_V}(d_{V,1}, \dots, d_{V,m}) \end{bmatrix}^T, \quad k_V \gg m. \quad (4.17)$$

In other words, for an input feature space with m dimensions the transformation $\mathbb{F}_V(d_{V,1}, \dots, d_{V,m})$ generates a k_V dimensional feature space with $k_V \gg m$, where each dimension i of the new feature space corresponds to the function $f_{V,i}(d_{V,1}, \dots, d_{V,m})$. The optimal number of such functions will be selected by the MDL principle and the method to estimate the functions is defined next.

4.4.1 Visual Features Pre-Processing

The feature processing step normalises the features and creates smaller-dimensional subspaces from the original feature-spaces. The low-level visual features that we use in our implementation are:

- **Marginal HSV distribution moments:** this 12 dimensional colour feature captures the 4 central moments of each colour component distribution. I use 3 subspaces corresponding to the 3 colour components with 4 dimensions each subspace.
- **Gabor texture:** this 16 dimensional texture feature captures the frequency response (mean and variance) of a bank of filters at different scales and orientations. I use 8 subspaces corresponding to each filter response of 2 dimensions each.
- **Tamura texture:** this 3 dimensional texture feature is composed of the image's coarseness, contrast and directionality.

I tiled the images in 3 by 3 parts before extracting the low-level features. This has two advantages: it adds some locality information and it greatly increases the amount of data used to learn the generic codebook.

4.4.2 Visual Transformation: Hierarchical EM

The original visual feature vector $d_V = (d_{V,1}, \dots, d_{V,m})$ is composed of several low-level visual features with a total of m dimensions. These m dimensions span the J visual feature types (e.g., marginal HSV colour moments, Gabor filters and Tamura), i.e. the sum of the number of dimensions of each one of the J visual feature space equals m . This implies that each visual feature type j is transformed individually by the corresponding $\mathbb{F}_{V,j}(d_{V,j})$ and the output is

concatenated into the vector

$$\mathbf{F}_V(d_V) = \begin{bmatrix} \mathbf{F}_{V,1}(d_{V,1}) \\ \vdots \\ \mathbf{F}_{V,j}(d_{V,j}) \end{bmatrix}^T, \quad (4.18)$$

where the dimensionality of the final \mathbf{F}_V transformation is the sum of the dimensionality of each individual visual feature space transformation $\mathbf{F}_{V,j}$, i.e.,

$$k_V = k_{V,1} + \dots + k_{V,j} + \dots + k_{V,J}. \quad (4.19)$$

The form of visual feature space transformations $\mathbf{F}_{V,j}$ is based on Gaussian mixture density models. The components of a GMM capture the different modes of the problem's data. I propose to use each component as a dimension of the optimal feature space where modes are split and well separated thereby creating a feature space where keywords can be modelled with a simple and low cost algorithm.

The transformations are defined under the assumption that subspaces are independent. This allows us to process each visual feature subspace j individually and model it as a Gaussian mixture model (GMM)

$$p(d_V) = p(d_V | \theta_j) = \sum_{m=1}^{k_{V,j}} \alpha_{m,j} p(d_V | \mu_{m,j}, \sigma_{m,j}^2), \quad (4.20)$$

where d_V is the low-level feature vector, θ_j represents the set of parameters of the model of the j visual feature subspace: the number $k_{V,j}$ of Gaussians components, the complete set of model parameters with means $\mu_{m,j}$, covariances $\sigma_{m,j}^2$, and component priors $\alpha_{m,j}$. The component priors have the convexity constraint $\alpha_{1,j}, \dots, \alpha_{k_{V,j},j} \geq 0$ and $\sum_{m=1}^{k_{V,j}} \alpha_{m,j} = 1$. Thus, for each visual feature space j , we have the Gaussian mixture model with $k_{V,j}$ components which now defines the transformation,

$$\mathbf{F}_{V,j}(d_V) = \begin{bmatrix} \alpha_{1,j} p(d_V | \mu_{1,j}, \sigma_{1,j}^2) \\ \vdots \\ \alpha_{k_{V,j},j} p(d_V | \mu_{k_{V,j},j}, \sigma_{k_{V,j},j}^2) \end{bmatrix}^T, \quad (4.21)$$

where each dimension corresponds to a component of the mixture model. The critical question that arises from the above expression is that one does not know the optimal complexity of the GMM in advance. The complexity is equivalent to the number of parameters, which in our case is

proportional to the number of mixture components $k_{V,j}$:

$$npars_j = k_{V,j} + \dim_j \cdot k_{V,j} + k_{V,j} \frac{\dim_j \cdot (\dim_j + 1)}{2}, \quad (4.22)$$

where \dim_j is the dimensionality of the visual subspace j . Note the relation between this equation and Equation (4.14). To address the problem of finding the ideal complexity we implemented a hierarchical EM algorithm that starts with a large number of components and progressively creates different GMM models with a decreasing number of components. For example, if it starts with 10 random components the EM will fit those 10 GMM components, store that model, deletes the weakest component and restarts the fitting with the previously 9 fitted components that will compensate the deleted component. The process is repeated until one component remains. In the end the algorithm generated 10 mixtures that are then assessed with the MDL criterion and the best one is selected. The implemented hierarchical EM adopts several other strategies that we will describe next.

Implementation Details

The hierarchical EM algorithm was implemented in C++ and it is based on the one proposed by Figueiredo and Jain (2002): it follows the component-wise EM algorithm with embedded component elimination. Figure 4.3 presents its pseudo-code; more details can be found in (Figueiredo and Jain 2002). The mixture fitting algorithm presents a series of strategies that avoids some of the EM algorithm’s drawbacks: sensitivity to initialization, possible convergence to the boundary of the parameter space and the estimation of different feature importance.

The algorithm starts with a number of components that is much larger than the real number and gradually eliminates the components that start to get few support data (singularities). This avoids the initialization problem of EM since the algorithm only produce mixtures with components that have enough support data. Component stability is checked by assessing its determinant (close to singularity) and its prior (few support data). If one of these two conditions is not met, we delete the component and continue with the remaining ones. This strategy can cause a problem when the initial number of components is too large: no component receives enough initial support causing the deletion of all components. To avoid this situation, component parameters are updated sequentially and not simultaneously as in standard EM. That is: first update component 1 parameters (μ_1, σ_1^2) , then recompute all posteriors, update component 2 parameters (μ_2, σ_2^2) , recompute all posteriors, and so on.

After finding a good fit for a GMM with k components, the algorithm deletes the weakest component and restarts itself with $k - 1$ Gaussians and repeats the process until a minimum

number of components is reached. Each fitted GMM is stored and in the end the set of fitted models describe the feature subspace at different levels of granularities.

The hierarchical EM algorithm for Gaussian mixture models addresses the objective of finding the optimal feature space by (1) creating transformations with different complexities and (2) splitting data modes into different space dimensions, hence enabling the application of low-cost keyword modelling algorithms.

```

Input: data, k_max, k_min, threshold, MinPrior, MinVolume

for (k = 1; k < k_max; k++) {
    GMM[k].Initialize(data);

    // This cycle fits several mixture models
    while (k_max > k_min) {

        // This cycle fits one mixture model
        do {
            for (k = 1; k < k_max; k++) {
                // Maximization-Step
                GMM[k].UpdateMean();
                GMM[k].UpdateCovariance();
                GMM[k].UpdatePrior();

                // Check for singularities and small components
                if ((Det(GMM[k].Covariance()) < MinVolume) ||
                    (GMM[k].Prior < MinPrior)) {

                    GMM[k].DeleteComponent();
                    k_max = k_max - 1;
                }

                // Expectation-Step
                UpdatePosteriors();
            }
            old_llk = llk;
            llk = LogLikelihood();

        } while (threshold > (llk - old_llk));

        // Store the fitted mixture model
        HierarchyOfGMM.Push(GMM);

        // Restart the algorithm without the smallest component
        GMM.DeleteWeakestComponent();
        k_max = k_max - 1;
    }

Output: HierarchyOfGMM

```

Figure 4.3. Hierarchical EM algorithm.

4.4.3 Experiments

Experiments assessed the behaviour of the hierarchical EM algorithm on a real world

photographic image collection. The collection is a 4,500 images subset of the widely used Corel CDs Stock Photos. More details regarding this collection are provided in Chapter 2. The visual features used in these experiments are the Gabor texture features, the Tamura texture features and the marginal HSV colour moments as described in Section 4.4.1.

The evolution of the model likelihood and complexity with a decreasing number of components are the two most important characteristics of the hierarchical EM that I wish to study. The algorithm is applied to individual visual feature subspaces. Each GMM model starts with $k_{V,j} = 200$ Gaussians, and the algorithm fits models with a decreasing number of components until a minimum number of Gaussians of 1.

One of the assumptions of the minimum description length principle is that the number of samples is infinite. Thus, to increase the accuracy of the MDL criterion we created 3 by 3 tiles of the training images. This increased the number of training samples by a factor of 9, which greatly improves the quality of the produced GMMs because of the existence of more data to support the model parameters.

The inclusion of all tiles also brings another advantage: it allows algorithms to explore the correlation between different concepts present in different tiles. For example, because most pictures of *jets* are taken with a *jet* on the central tile and *sky* on the surrounding tiles, this constitutes a strong correlation that algorithms should capture.

4.4.4 Results and Discussion

An advantage of the chosen algorithm to find the optimal transformation is its natural ability to generate a series of transformations with different levels of complexities. This allows assessing different GMMs with respect to the trade-off between decreasing levels of granularity and their fit to the data likelihood.

Figure 4.4 illustrates the output of a GMM model fitting to the output of one Gabor filter. The minimum description length curve (blue line) shows the trade-off between the models complexity (green line) and the models likelihood (red line). Note that we are actually plotting $-\log\text{-likelihood}$ for better visualization and comparison. The models likelihood curve is quite stable for models with a large number of components (above 40). On the other extreme of the curve one can see that models with fewer than 40 components the likelihood start to exhibit a poorer performance. The small glitches in the likelihood curve are the result of component deletion from a particularly good fit (more noticeable between 10 and 20 components). This effect is more visible when a component has been deleted from a model with a low number of components because the remaining ones are not enough to cover the data that was supporting the deleted one.

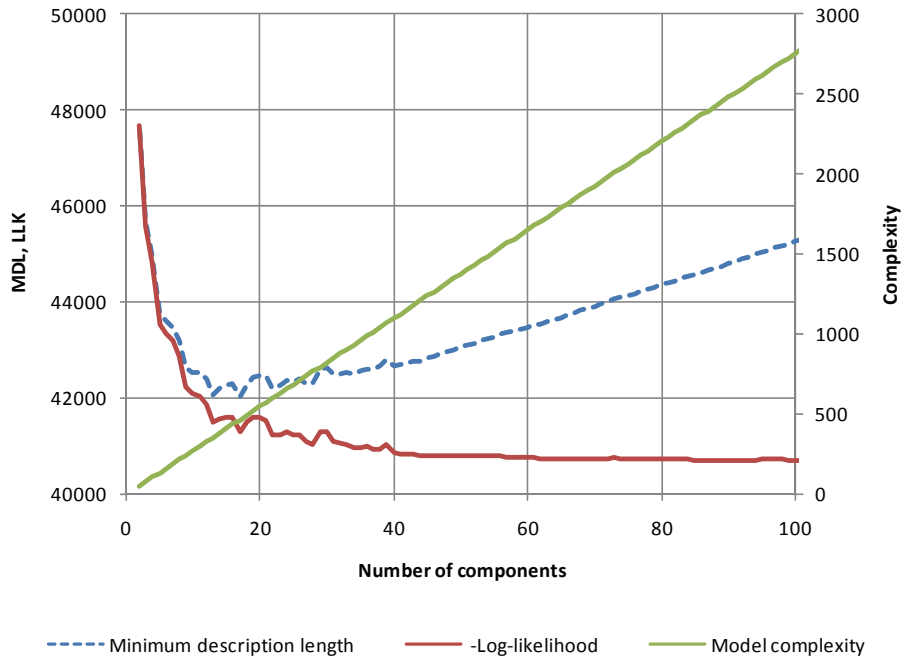


Figure 4.4. Model selection for the Gabor filters features (Corel5000).

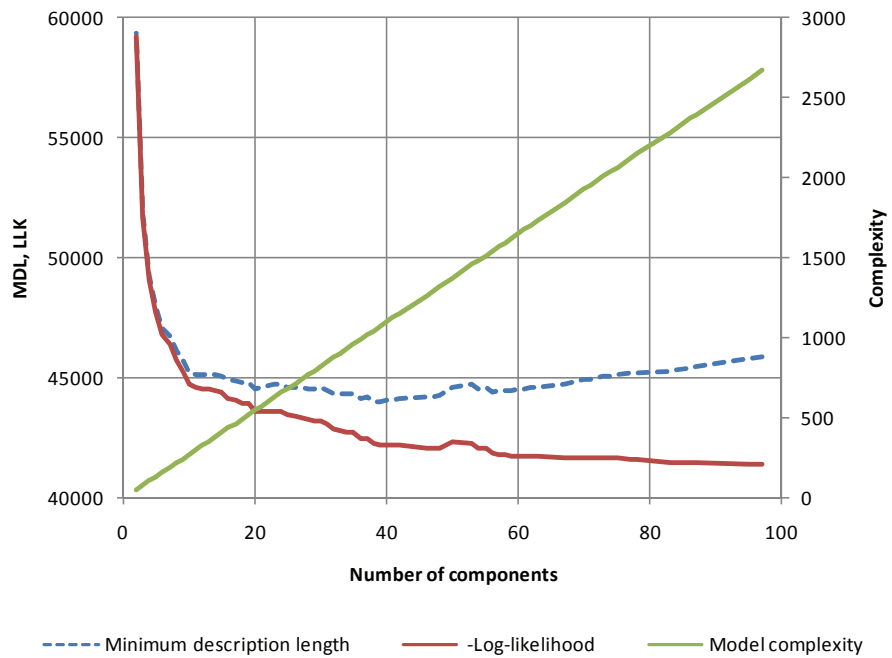


Figure 4.5. Model selection for the Tamura features (Corel5000).

The model complexity curve shows the penalty increasing linearly with the number of components according to Equation (4.22). The most important curve of this graph is the minimum description length curve. At the beginning it closely follows the likelihood curve because the complexity cost is low. As the model complexity increases the model likelihood also becomes better but no longer at the same rate as initially (less than 10 components). This causes the model penalty to take a bigger part in the MDL formula, and after 20 components the MDL criterion indicates that those models are not better than previous ones. Thus, according to the MDL criterion the optimal transformation for this Gabor filter is the model with 18 components.

The selection of the transformation of the Tamura visual texture features is illustrated in Figure 4.5. The behaviour is the same as for the Gabor features with the only difference that the change from the descending part of the MDL curve to the ascending part is not so pronounced. This indicates that the optimal model, $k_{V,j} = 39$, is not so distinct from the neighbouring models with $k_{V,j}$ between 30 and 50.

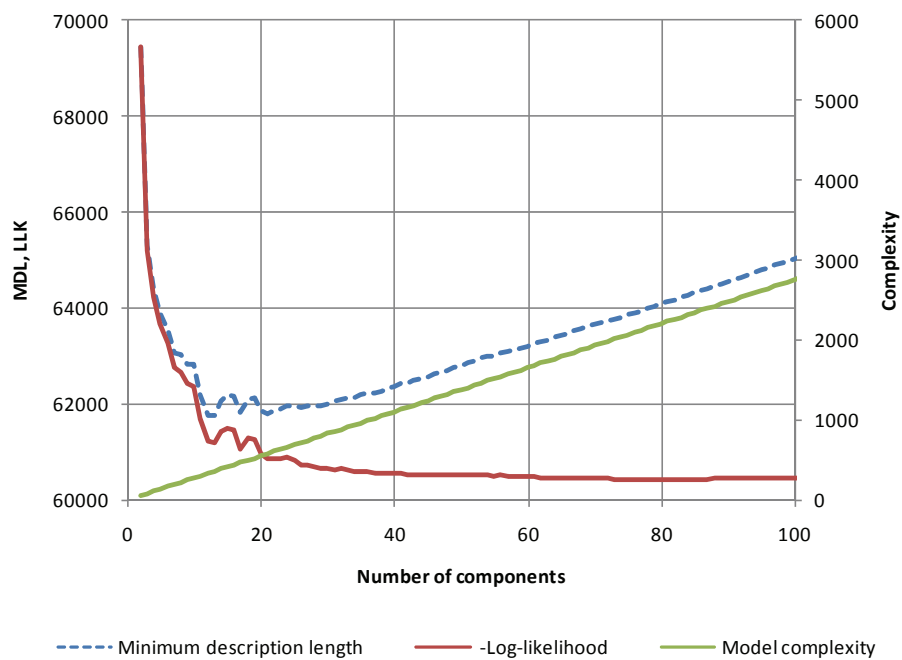


Figure 4.6. Model selection for the marginal moments of HSV colour histogram features (Core15000).

Finally, Figure 4.6 illustrates the optimal transformation selection experiments for a colour channel of the marginal HSV colour moments histograms. The behaviour is again similar to the previous ones and the optimal model, $k_{V,j} = 12$, is quite distinct from the surrounding neighbours. Note that the likelihood curve glitches are again present in this feature space which is

an indication that the GMMs are well fitted to the data with a low number of components and that a deletion of a component leaves uncovered data causing the likelihood jitter.

4.5 Sparse Spaces Transformations

Text features are high-dimensional sparse data, which pose some difficulties to parametric generative models because each parameter receives little data support. In discriminative models one observes over-fitting effects because the data representation might be too optimistic by leaving out a lot of the underlying data structure information. High-dimensional sparse data must be compressed into a lower dimensional space to ease the application of generative models. This optimal data representation is achieved with a transformation function defined as

$$F_T(d_{T,1}, \dots, d_{T,n}) = \begin{bmatrix} f_{T,1}(d_{T,1}, \dots, d_{T,n}) \\ \vdots \\ f_{T,k_T}(d_{T,1}, \dots, d_{T,n}) \end{bmatrix}^T, \quad k_T \ll n, \quad (4.23)$$

where n is the number of dimensions of the original sparse space, and k_T is the number of dimensions of the resulting optimal feature space.

In other words, the sparse spaces transformation $F_T(d_{T,1}, \dots, d_{T,n})$ receives as input a feature space with n dimensions and generates a k_T dimensional feature space, where each dimension i of the new optimal feature space corresponds to the function $f_{T,i}(d_{T,1}, \dots, d_{T,n})$. The optimal number of such functions will be selected by the MDL principle, and the method to estimate the functions is defined next.

4.5.1 Text Feature Pre-Processing

The text part of a document is represented by the feature vector $d_T = (d_{T,1}, \dots, d_{T,n})$ obtained from the text corpus of each document by applying several standard text processing techniques (Yang 1999): stop words are first removed to eliminate redundant information, and rare words are also removed to avoid over-fitting (Joachims 1998). After this, the Porter stemmer (Porter 1980) reduces words to their morphological root, which we call *term*. Finally, we discard the term sequence information and use a bag-of-words approach.

These text pre-processing techniques result in a feature vector $d_T = (d_{T,1}, \dots, d_{T,n})$, where each $d_{T,i}$ is the number of occurrences of term t_i in document d .

4.5.2 Text Codebook by Feature Selection

To reduce the number of dimensions in a sparse feature space we rank terms t_1, \dots, t_n by their importance to the modelling task and select the most important ones. The information gain criterion ranks the text terms by their importance, and the number of text terms is selected by the minimum description length. The criterion to rank the terms is the average mutual information technique, also referred to as information gain (Yang 1999), expressed as

$$\text{IG}(t_i) = \frac{1}{L} \sum_{j=1}^L \text{MU}(y_j, t_i), \quad (4.24)$$

where t_i is term i , and y_j indicates the presence of keyword w_j . The information gain criterion is the average of the mutual information between each term and all keywords. Thus, one can see it as the mutual information between a term t_i and the keyword vocabulary.

The mutual information criterion assess the common entropy between a keyword entropy $H(y_j)$ and the keyword entropy given a term t_i , $H(y_j | t_i)$. Formally the mutual information criterion is defined as

$$\text{MU}(y_j, t_i) = \sum_{y_j \in \{0,1\}} \sum_{d_{T,i}} p(y_j, d_{T,i}) \log \frac{p(y_j, d_{T,i})}{p(y_j) p(d_{T,i})}, \quad (4.25)$$

where $d_{T,i}$ is the number of occurrences of term t_i in document d . Yang and Pedersen (1997) and Forman (2003) have shown experimentally that this is one of the best criteria for feature selection. A document d is then represented by k_T text terms as the mixture

$$p(d) = \sum_{i=1}^{k_T} \alpha_i p(t_i | d) = \sum_{i=1}^{k_T} \alpha_i \frac{d_{T,i}}{|d|}, \quad (4.26)$$

where $d_{T,i}$ is number of occurrences of term t_i in document d . The parameters of the above mixture are the priors α_i of corresponding to term t_i . This results in a total number of parameters

$$npars = k_T. \quad (4.27)$$

A list of models is constructed by progressively adding terms to each model according to the order established by the information gain criterion. In this particular case of sparse text features the complexity of the transformation is equivalent to the number k_T of text terms. The application of the MDL criterion in Equation (4.14) is now straightforward.

Finally, terms are weighted by their inverse document frequency, resulting in the feature space

transformation function

$$f_{T,i}(d_T) = -d_{T,r(i)} \cdot \log \left(\frac{N}{\text{DF}(d_{T,r(i)})} \right), \quad (4.28)$$

where N is the number of documents in the collection, $\text{DF}(d_{T,i})$ is the number of documents containing the term t_i , and $r(i)$ is a permutation function that returns the i^{th} text term of the information gain rank.

4.5.3 Experiments

Experiments assessed the behaviour of the information gain criterion on the Reuters news collection described in Chapter 2. The text corpus was processed as described in Section 4.5.1 to obtain the text terms, and models are constructed by adding terms to the model according to the information gain rank.

4.5.4 Results and Discussion

The evolution of the model likelihood and complexity with an increasing number of terms is again the most important characteristic that we wish to study. Figure 4.7 illustrates the model likelihood (red line) versus the model complexity (green line) and the minimum description length criterion as a measure of their trade-off. Note that the graph is actually showing the *-log-likelihood* for easier visualization and comparison.

Figure 4.7 illustrates the improving likelihood as new terms are added to the feature space. The curve smoothness observed in this graph is due to the scale of the x-axis (100 times greater than in the images case) and to the fact that neighbouring terms have similar information value.

The problem of selecting the dimensionality of the optimal feature space is again answered by the minimum description length criterion that selects a feature space with 972 dimensions. It is interesting to notice that the MDL selects a low dimensionality reflecting a model with lower complexity than others with better likelihood but higher complexity. Note that if we had more samples (in this dataset the number of samples is limited to 7,770) we would be able to select a more complex model (remember that the MDL criterion assumes an infinite number of samples).

Moreover, information gain is a feature selection method that ranks terms by their discriminative characteristics and does not actually try to faithfully replicate the data characteristics. This is in contrast with the hierarchical EM method used for the dense feature spaces that is a pure generative approach. Hence, when adding new terms to the optimal feature space, we are directly affecting the classification performance.

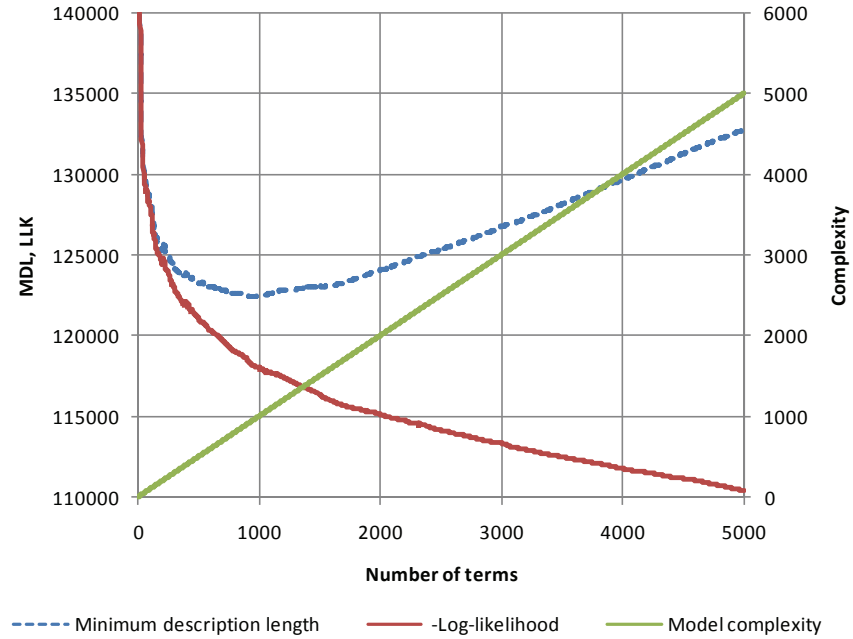


Figure 4.7. Model selection for the bag-of-words features (Reuters).

4.6 Conclusions and Future Work

This Chapter proposed a probabilistic framework aimed at extracting the semantics of multimedia information. The probabilistic framework, summarized in the expression

$$p\left(y_t^j = 1 \mid \mathbb{F}\left(d_T^j, d_V^j\right), \beta_t\right), \quad (4.29)$$

is divided in two parts: the support of heterogeneous types of data through the feature space transformation $\mathbb{F}\left(d_T^j, d_V^j\right)$ and keyword models β_t . The support of heterogeneous types of data, the main topic of this Chapter, is one of the central points of a true multimedia information retrieval system. We looked at a list of requirements to guide the design of the feature space transformations. A distinction was made between the types of multimedia feature spaces as sparse feature spaces and dense feature spaces. In sparse spaces most dimensions of a feature vector are zero and in dense spaces most dimensions are non-zero and they have a high cross-interference between classes.

For dense spaces, we proposed a hierarchical EM algorithm as the feature space transformation $\mathbb{F}_V\left(d_V^j\right)$. The transformation uses the components of a Gaussian mixture model as dimensions of the optimal feature space. The optimal complexity of the mixture model is selected by the MDL

criterion. For sparse spaces, we proposed the average mutual information criterion as the feature space transformation $F_T(d_T^j)$. The transformation ranks terms by their relevance and the optimal feature space is obtained by selecting the optimal number of terms with the MDL criterion.

Experiments showed how the minimum description length criterion selects the optimal feature space transformation by assessing the trade-off between model likelihood and model complexity. The next chapter will show how the MDL criterion, a completely unsupervised criterion, can actually select an optimal (or close to optimal) multi-modal feature space.

4.6.1 Future Work

The presented research triggered some ideas that we wish to pursue in the future:

- **Text transformation:** the information gain criterion depends on keyword class information, which contrasts with visual feature transformations that are completely independent of this class information. Thus, one of the items that we plan to include in this framework in the future is a text clustering technique that does not discard text terms.
- **High-dimensional indexing methods:** the fitting of hierarchical GMM models create a structured representation of data similar to the ones used in high-dimensional indexing methods. We would like to investigate the applicability of this hierarchy to improve the index search efficiency and computational complexity.

Keyword Models

5.1 Introduction

Modelling keywords in terms of multimedia information is the main objective of the first part of this thesis. Keywords are present in multimedia documents according to complex patterns that reflect their dependence and correlations. Different probability distributions can be applied to capture this information, also Bayesian networks can be used to define complex distributions that try to represent complex keyword interactions. This thesis opted to assume nothing about keyword interactions, and we define keywords as Bernoulli random variables with

$$p(y_t = 1) = 1 - p(y_t = 0) = \frac{|\mathcal{D}_{w_t}|}{|\mathcal{D}|}, \quad (5.1)$$

where y_t is a particular keyword, $|\mathcal{D}|$ is the size of the training collection and $|\mathcal{D}_{w_t}|$ is the number of documents in the training collection containing keyword w_t . In the previous chapter we proposed a probabilistic framework

$$p(y_t | F(d), \beta_t), \quad y_t = \{0, 1\}, \quad (5.2)$$

where $F(d)$ is a visual and text data transformation that creates a unique multi-modal feature space, and a keyword w_t is represented in that feature space by a model β_t . We will ignore the feature type and use a plain vector to represent the low-level features of a document as

$$F(d^j) = (F_T(d_T), F_V(d_V)) = (f_1^j, \dots, f_M^j). \quad (5.3)$$

One of the goals of the proposed $F(d)$ transformation is the creation of an optimal feature

space, where simple and scalable keyword models β_t can be used. This chapter will propose the application of linear models to address this particular problem. The setting is a typical supervised learning problem, where documents are labelled with the keywords that are present in that document. Thus, we define

$$y^j = (y_1^j, \dots, y_L^j), \quad (5.4)$$

as the binary vector of keyword annotations of document j , where each y_t^j indicates the presence of keyword w_t in document j if $y_t^j = 1$. Note that a perfect classifier would have $(y - d_W) = 0$ on a new document. The annotations vector y^j is used to estimate keyword models and to test the effectiveness of the computed models.

5.2 Keyword Baseline Models

The first linear models that we shall present in this section are simple but effective models that can be applied in the multi-modal feature space (Magalhães and Rügger 2007a). The advantage of both Rocchio classifier and naïve Bayes classifier is that they can be computed analytically.

5.2.1 Rocchio Classifier

Rocchio classifier was initially proposed as a relevance feedback algorithm to compute a query vector from a small set of positive and negative examples (Rocchio 1971). It can also be used for categorization tasks, e.g., (Joachims 1997): a keyword w_t is represented as a vector β_t in the multi-modal space, and the closer a document is to this vector the higher is the similarity between the document and the keyword. A keyword vector β_t is computed as the average of the vectors of both relevant documents $\{\mathcal{D}_{w_t}\}$ and non-relevant documents $\{\mathcal{D} \setminus \mathcal{D}_{w_t}\}$,

$$\beta_t = \frac{1}{|\mathcal{D}_{w_t}|} \sum_{d \in \mathcal{D}_{w_t}} \frac{F(d)}{\|F(d)\|} - \frac{1}{|\mathcal{D} \setminus \mathcal{D}_{w_t}|} \sum_{d \in \mathcal{D} \setminus \mathcal{D}_{w_t}} \frac{F(d)}{\|F(d)\|}. \quad (5.5)$$

For retrieval scenarios, documents are ranked according to their proximity to the keyword vector. The cosine similarity measure has already proven to perform quite well in high-dimensional spaces. Since the cosine function is limited to the interval $[-1; 1]$ one can define the probability of observing a keyword w_t in a particular document d as a function of the cosine of the angle between the keyword vector β_t and the document vector, i.e.,

$$p(w_t | d) = \frac{1}{2} + \frac{1}{2} \cos(\beta_t, F(d)), \quad (5.6)$$

where the $\cos(\beta_t, F(d))$ is computed as

$$\cos(\beta_t, F(d)) = \frac{\beta_t}{\|\beta_t\|} \cdot \frac{F(d)}{\|F(d)\|} = \frac{\sum_{i=1}^T \beta_{t,i} \cdot f_i}{\sqrt{\sum_{i=1}^M (\beta_{t,i})^2} \cdot \sqrt{\sum_{i=1}^M (f_i)^2}}. \quad (5.7)$$

The Rocchio classifier is a simple classifier that has been widely used in the area of text information retrieval and, as we have shown, can also be applied to semantic-multimedia information retrieval. Moreover, this classifier is particularly useful for online learning scenarios and other interactive applications where the models need to be updated on-the-fly or the number of training examples are limited.

5.2.2 Naïve Bayes Model

The naïve Bayes classifier assumes independence between feature dimensions and is the result of the direct application of Bayes's law to classification tasks:

$$p(y_t = 1 | d) = \frac{p(y_t = 1) p(d = f_1, \dots, f_M | y_t = 1)}{p(d)} \quad (5.8)$$

The assumption that features f_i are independent of each other in a document can be modelled by several different independent probability distributions. A distribution is chosen according to some constraints that we put on the independence assumptions. For example, if we assume that features f_i can be modelled as the simple presence or absence in a document then we consider a binomial distribution. If we assume that features f_i can be modelled as a discrete value to indicate the presence confidence in a document then we consider a multinomial distribution, see (McCallum and Nigam 1998). The binomial distribution over features f_i would be too limiting; the multinomial distribution over features f_i offers greater granularity to represent a feature value.

In the multi-modal feature space features are continuous and not discrete. Thus, we need to define $N_{f_i|d}$ as the *count* of the feature f_i in a given document d . To satisfy the multinomial distribution this variable needs to be an integer and we approximate it as

$$N_{f_i|d} = \lfloor p(f_i | d) \cdot M \rfloor. \quad (5.9)$$

Note that for high-dimensional feature spaces, M is quite large allowing us to round $N_{f_i|d}$ to an integer with minor loss of accuracy. Given this, the probability of a document d given a keyword w_t is expressed as a multinomial over all feature space dimensions:

$$p(d | y_t = 1) = p(|d|) |d|! \prod_{i=1}^M \frac{p(f_i | y_t = 1)^{N_{f_i|d}}}{N_{f_i|d}!} \quad (5.10)$$

When plugging the multinomial distribution into expression (5.8) the term $1/N_{f_i|d}!$ is cancelled. Since all documents have the same length, the constants $|d|!$ and $P(|d|)$ can be dropped from the equation. This leaves us with the proportionality relation

$$p(d | y_t = 1) \propto \prod_{i=1}^M p(f_i | y_t = 1)^{N_{f_i|d}}. \quad (5.11)$$

Now, we are left with the task of computing the probability of feature f_i for a given keyword w_t :

$$p(f_i | y_t = 1) = \frac{\sum_{d \in \mathcal{D}_{w_t}} f_i(d)}{\sum_{d \in \mathcal{D}} f_i(d)} \quad (5.12)$$

Finally, the complete expression of the naïve Bayes model assuming a multinomial behaviour of features f_i can be written as:

$$p(y_t = 1 | d) = \frac{p(y_t = 1) \prod_{i=1}^M p(f_i | y_t = 1)^{N_{f_i|d}}}{\sum_{y_j \in \{0,1\}} p(y_j) \prod_{i=1}^M p(f_i | y_j)^{N_{f_i|d}}} \quad (5.13)$$

This results in the following keyword models

$$\beta_{t,i} = p(f_i | y_t = 1), \quad i = 1, \dots, M. \quad (5.14)$$

In retrieval scenarios, documents are ranked according to their probability for the queried category. In classification scenarios, documents are labelled with the arguments that maximize the expression

$$\max_{t \in \{1, \dots, L\}} p(y_t | d). \quad (5.15)$$

Alternatively, one can compute the log-odds and classify a document with the keywords that have a value greater than zero:

$$\log \frac{p(w_j = 1 | d)}{p(w_j = 0 | d)} = \log \frac{p(y_t = 1)}{p(y_t = 0)} + M \sum_{i=1}^M p(f_i | d) \log \frac{p(f_i | y_t = 1)}{p(f_i | y_t = 0)} \quad (5.16)$$

Formulating naïve Bayes in log-odds space has two advantages: it shows that naïve Bayes is a linear model and avoids decision thresholds in multi-categorization problems. In this case the keyword models become

$$\beta_{t,i} = \log \frac{p(f_i | y_t = 1)}{p(f_i | y_t = 0)}, \quad i = 1, \dots, M. \quad (5.17)$$

5.3 Keywords as Logistic Regression Models

Logistic regression is a statistical learning technique that has been applied to a great variety of fields, e.g., natural language processing (Berger, Pietra and Pietra 1996), text classification (Nigam, Lafferty and McCallum 1999), and image annotation (Jeon and Manmatha 2004). In this section we employ a binomial logistic model to represent keywords in the multi-modal feature space. The expression of the binomial logistic regression is

$$p(y_t = 1 | F(d), \beta_t) = \frac{1}{1 + \exp(\beta_t \cdot F(d))} \quad (5.18)$$

and

$$p(y_t = 0 | F(d), \beta_t) = \frac{\exp(\beta_t \cdot F(d))}{1 + \exp(\beta_t \cdot F(d))}. \quad (5.19)$$

The logistic regression model is also a linear model, which makes it a scalable and efficient solution for modelling keywords. It can be easily shown that logistic regression is a linear model by computing the log-odds

$$\log \frac{p(y_t = 1 | F(d^j), \beta_t)}{p(y_t = 0 | F(d^j), \beta_t)} > 0, \quad (5.20)$$

as we did for the naïve Bayes classifier. If the inequality is true then the keyword is deemed to be present in the document. Expanding this equation we get

$$\beta_t F(d^j) = \beta_{t,0} + \beta_{t,1} f_1^j + \dots + \beta_{t,M} f_M^j > 0 \quad (5.21)$$

that shows the linear relationship between the regression coefficients β_t and the multi-modal features $F(d)$. Figure 4.1 shows the form of the binomial logistic regression function.

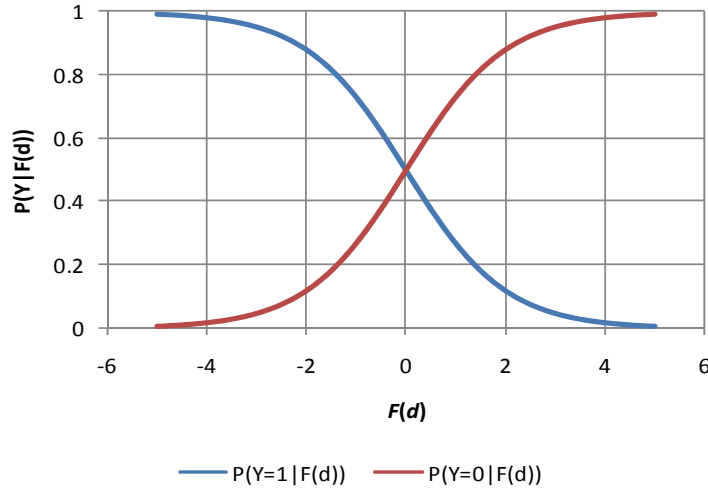


Figure 5.1. Form of the binomial logistic model.

The theory of Generalized Linear Models also shows how to derive the logistic regression expression from a point of view of pure linear models and without making use of the log-odds as we did here. I shall develop this later in this chapter.

5.3.1 Regularization

As discussed by Nigan, Lafferty and McCallum (1999) and Chen and Rosenfeld (1999), logistic regression may suffer from over-fitting. This is usually because features are high-dimensional and sparse meaning that the regression coefficients can easily push the model density towards some particular training data points. Zhang and Oles (2001) have also presented a study on the effect of different types of regularization on logistic regression. Their results indicate that with the adequate cost function (regularization), precision results are comparable to SVMs with the advantage of rendering a probabilistic density model.

An efficient and well known method of tackling over-fitting is to set a prior on the regression coefficients. As suggested by Nigan, Lafferty and McCallum (1999) and Chen and Rosenfeld (1999) I use a Gaussian prior \mathcal{N}_ξ for the regression coefficients,

$$\beta_* \sim \mathcal{N}_\xi(\mu_\xi, \sigma_\xi^2) \quad (5.22)$$

with mean $\mu_\xi = 0$ and σ_ξ^2 variance. The Gaussian prior imposes a cost on models β_* with large norms thus preventing optimization procedures from creating models that depend too much on a

single feature space dimension. When introducing the Gaussian prior in the keyword model expression we obtain

$$p(y_t = 1 | d, \beta_t, \sigma_\xi^2) = p(y_t = 1 | d, \beta_t) p(\beta_t | \sigma_\xi^2), \quad (5.23)$$

which we will now use in the maximum likelihood estimation. We will drop the variance σ_ξ^2 of the Gaussian prior in our notation.

5.3.2 Maximum Likelihood Estimation

The log-likelihood function computes the sum of the log of the errors of each document in the collection \mathcal{D} :

$$l(\beta_t | \mathcal{D}) = \sum_{j \in \mathcal{D}} \log(p(y_t^j | F(d^j), \beta_t) p(\beta_t)) \quad (5.24)$$

For each keyword model the likelihood function tells us how well the model and those parameters represent the data. The model is estimated by finding the minimum of the likelihood function by taking the regression coefficients as variables:

$$\beta_t = \min_{\beta} l(\beta | \mathcal{D}) \quad (5.25)$$

For models where the solution can be found analytically, the computation of the regression coefficients is straightforward. In cases, where the analytical solution is not available typical numerical optimization algorithms are adequate.

The regression coefficients need to be found by a numerical optimization algorithm that iteratively approaches a solution corresponding to a local minimum of the log-likelihood function. To find the minimum of the log-likelihood function $l(\beta)$ with respect to β , I use the Newton-Raphson algorithm:

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 l(\beta^{old})}{\partial \beta \partial \beta^T} \right)^{-1} \left(\frac{\partial l(\beta^{old})}{\partial \beta} \right) \quad (5.26)$$

The first-order derivative matrix is a vector with M elements corresponding to the dimension of the space resulting from the application of $F(d)$ to the original data. The second-order derivative, the Hessian matrix, is a square-matrix with $M \times M$ components. The Hessian matrix imposes a high computational complexity (both in time and space) on the parameter estimation

algorithm. In multimedia information retrieval we use feature spaces with thousands of dimensions, meaning that the processing of the Hessian matrix is computationally too costly. For these reasons, we must use algorithms that are more suitable for such a large-scale problem.

5.3.3 Large-Scale Model Computation

When applying the Newton-Raphson algorithm to high-dimensional data the Hessian matrix often cannot be computed at a reasonable cost because it is too large and dense. Large scale Quasi-Newton methods are an adequate solution for our problem: instead of storing and computing the full Hessian matrix, these methods store a few vectors that represent approximations implicitly made in previous iterations of the algorithm. The L-BFGS algorithm (limited-memory Broyden-Fletcher-Goldfarb-Shanno) is one of such algorithms, see (Liu and Nocedal 1989a) for details: “*The main idea of this method is to use curvature information from only the most recent iterations to construct the Hessian approximation. Curvature information from earlier iterations, which is less likely to be relevant to the actual behaviour of the Hessian at the current iteration, is discarded in the interest of saving storage.*”

The L-BFGS algorithm iteratively evaluates the log-likelihood function and its gradient, and updates the regression coefficients and the Hessian approximation. For the binomial logistic regression the log-likelihood function is

$$l(\beta_t) = \sum_{d^j \in \mathcal{D}} \left(y_t^j \beta_t F(d^j) - \log \left(1 + \exp \left(\beta_t F(d^j) \right) \right) \right) - \lambda \beta_t^2, \quad (5.27)$$

$$\lambda = \frac{1}{2\sigma_\xi^2},$$

where for each example d^j the variable y_t^j is 1 if the example contains the keyword w_t and 0 otherwise. $F(d^j)$ is the nonlinear space transformation of the document features. To minimize the log-likelihood we need to use the gradient information to find the β_t where the log-likelihood gradient is zero, i.e.,

$$\frac{\partial l(\beta_t)}{\partial \beta_t} = 0 = \sum_{d^j \in \mathcal{D}} F(d^j) \left(y_t^j - p \left(y_t^j = 1 \mid \beta_t, F(d^j) \right) \right) - \lambda \beta_t. \quad (5.28)$$

These two last equations are the binomial logistic regression functions that the L-BFGS algorithm evaluates on each iteration to compute the β_t regression coefficients.

We use the implementation provided by Liu and Nocedal (1989b) to estimate the parameters of both linear logistic models and log-linear models. It has been shown that L-BFGS is the best optimization procedure for both maximum entropy (Malouf 2002) and conditional random fields models (Sha and Pereira 2003). For more details on the limited-memory BFGS algorithm see

(Nocedal and Wright 1999).

5.4 Relationship to other Approaches

The proposed framework has several similarities with other approaches to statistical learning theory and pattern recognition. In this section we shall discuss and compare our probabilistic framework to other approaches that I consider to be the most similar.

5.4.1 Kernel Methods

Kernel methods, initially proposed by Aizerman, Braverman and Rozonoer (1964), use functions that allow learning algorithms to operate in a high-dimensional feature space without ever computing the vectors of the data in that space. Instead, one uses a kernel function to compute the inner products between vectors of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the high-dimensional vector.

These methods allow the application of linear classification algorithms to solve non-linear problems by transforming the original data into a higher-dimensional space, where the linear classifier can be applied.

The logic behind these methods is similar to our approach but with different motivations and ways of achieving it. While kernel methods use specific functions to transform data into the high-dimensional space we estimate the function from the data itself. Another difference is that we explicitly compute the vectors in the high-dimensional space. Note, however, that we cannot directly apply the kernel trick to our data because the concatenation of all features is already high-dimensional and quite heterogeneous.

5.4.2 pLSA

This family of algorithms merge the two steps that we defined and simultaneously optimize the creation of clusters and the cluster-class relation. However, usual pLSA do not define the number of clusters and leave this problem out of the formulation. If enough clusters are created, pLSA should obtain better results because the clustering phase is done with the objective of creating keyword-based clusters. This contrasts with the proposed framework that creates unsupervised clusters, and each keyword can have its own clusters if that improves the likelihood.

The implications of using pLSA to the task of multimedia semantic analysis is that existing indexes become invalid under several situations: when a new keyword is added to the vocabulary, all existing keyword models become invalid; when a single keyword needs to be updated with new training data, all existing keyword models become invalid as well. This effect is a consequence of

the joint modelling of keywords.

5.4.3 Generalized Linear Models

The classic linear model assumes a normal linear relationship between the input variables X and the model output for each keyword y_t :

$$E[y_t = 1 | X] = \beta_t X, \quad (5.29)$$

where β_t is the vector of regression coefficients of the keyword w_t . This linear relation is quite limited to model complex and non-linear processes as is the case at hand. Generalized linear models (McCullagh and Nelder 1989) extend this model by introducing a link function $g(\cdot) = \cdot$ to model non-linear relations between the input variables x and the model output y :

$$g(E[y_t = 1 | X]) = \beta_t X \quad (5.30)$$

The link function $g(\cdot) = \cdot$ can be any monotonic differentiable function that best represents the true relationship between the input variables and the model output. From the large number of possibilities for the link functions we used the logit function, Equation (5.18).

It is extremely rare that the true relationship between the input variables and the model output is actually linear. In these cases, basis functions are used to augment or replace the input variables by other variables that will turn the problem into a linear classification problem. In our framework this is the role of the feature space transformation $F(d^j) = (f_1^j, \dots, f_M^j)$. However, we have established a fully automated way of estimating these basis functions. Thus, the proposed probabilistic framework falls under the large category of Generalized Linear Models.

5.5 Evaluation

The presented algorithms were evaluated with a retrieval setting on the Reuters-21578 collection, on a subset of the Corel Stock Photo CDs (Duygulu et al. 2002) and on a subset of the TRECVID2006 development data.

5.5.1 Collections

The three collections used in this evaluation are described in more detail in Chapter 2.

Reuters-21578

This is a widely used text dataset which allows comparing our results with others in the literature. Each document is composed by a text corpus, a title (which we ignore), and labelled

categories. This dataset has several possible splits and we have used the *ModApte* split which contains 9,603 training documents and 3,299 test documents. This is the same evaluation setup used in several other experiments (Joachims 1998; Nigam, Lafferty and McCallum 1999; McCallum and Nigam 1998; Zhang and Oles 2001). Terms appearing less than 3 times were removed. Only labels with at least 1 document on the training set and the test set were considered leaving us with 90 labels. After these steps we ended with 7,770 labelled documents for training.

Corel Images

This dataset was compiled by Duygulu et al. (2002) from a set of COREL Stock Photo CDs. The dataset has some visually similar concepts (jet, plane, Boeing), and some concepts have a limited number examples (10 or less). In their seminal paper, the authors acknowledge that fact and ignored the classes with these problems. In this paper we use the same setup as in (Yavlinsky, Schofield and R ger 2005), (Carneiro and Vasconcelos 2005), (Jeon, Lavrenko and Manmatha 2003), (Lavrenko, Manmatha and Jeon 2003) and (Feng, Lavrenko and Manmatha 2004), which differs slightly from the one used in the dataset original paper, (Duygulu et al. 2002). The retrieval evaluation scenario consists of a training set of 4,500 images and a test set of 500 images. Each image is annotated with 1-5 keywords from a vocabulary of 371 keywords. Only keywords with at least 2 images in the test set and training set each were evaluated, which reduced the number of vocabulary to 179 keywords. Retrieval lists have the same length as the test set, i.e. 500 items.

TRECVID

To test the similarity ranking on a multi-modal data we used the TRECVID2006 data: since only the training set is completely labelled, we randomly split the training English videos into 23,709 training documents and 12,054 test documents. We considered each document to be a key-frame plus the ASR text within a window of 6 seconds around that key-frame. Key-frames are annotated with the standard vocabulary of 39 keywords provided by NIST.

5.5.2 Experiment Design

To evaluate the proposed framework we deployed a retrieval experiment for all collections listed in the previous section. The experiment methodology was as follows:

1. For a given algorithm and a given multi-modal feature space
 - a. For each keyword in the considered collection
 - i. Estimate the keyword model on the training set by applying a cross-validation with 5 folds and 10 value iterations, as suggested in (Kohavi

- 1995), to determine the ideal Gaussian prior variance σ_{ξ}^2
- ii. Compute the relevance of each test document
 - iii. Rank all test documents by their relevance for the considered keyword
 - iv. Use the collection relevance judgments to measure the retrieval effectiveness of the considered rank
- b. Repeat step a) for all keywords
 - c. Compute the mean average precision
2. Repeat for a different algorithm or multi-modal feature space

The above methodology was repeated for all linear models that we presented in this chapter and for different multi-modal feature spaces. We considered the Reuters-21578 collection, the Corel5000 collection, the ASR part of the TRECVID2006, the key-frames of the TRECVID2006 and both key-frames and text of the TRECVID2006 development data, which makes a total of five collections.

Feature Selection

The high dimensionality of the feature space could be reduced by applying a feature selection criterion to remove noisy dimensions. However, this introduces an unknown variable which is the ideal number of feature dimensions. These noisy dimensions are critical for numerical optimization procedures that exploit the small variations of each feature dimension. Standard logistic regression is highly affected by these noisy dimensions. The inclusion of a Gaussian prior allows to simultaneously estimate the keyword model and to *shrink* the weight of noisy features. Sparse linear models (SVMs and logistic regression with Laplacian prior) provide a better alternative by not including noisy feature dimensions. See (Hastie, Tibshirani and Friedman 2001), Section 3.5 for more details on feature selection and shrinkage.

5.5.3 Text-Only Models

The text-only models experiments on the Reuters-21578 collection evaluated the sparse data processing part of our framework. The optimal feature space was created with the average mutual information criterion as described in Chapter 4. All presented linear models were used in the evaluation.

Retrieval Effectiveness

Experiments in the Reuters dataset were evaluated with mean average precision, Figure 5.2, mean precision at 20, Figure 5.3, and interpolated precision-recall curves, Figure 5.4. All results were obtained with a 972 dimensional multi-modal feature space selected by the minimum description length criterion.

When comparing the naïve Bayes model to the logistic regression model, results confirm what one would expect: naïve Bayes performs much worse than logistic regression (24.3% MAP versus 49.0% MAP). However, it is a surprise to see that Rocchio classifier is actually comparable to logistic regression – it obtained 49.7%. This supports the hypothesis that Reuters data is structured in a single cluster shape. Another reason why the Rocchio classifier performs so well on this dataset is that from all three classifiers it is the one that uses the simplest assumptions about data (organized as a high-dimensional sphere). The implications are that it is less prone to over-fit on classes with few training examples, unlike logistic regression.

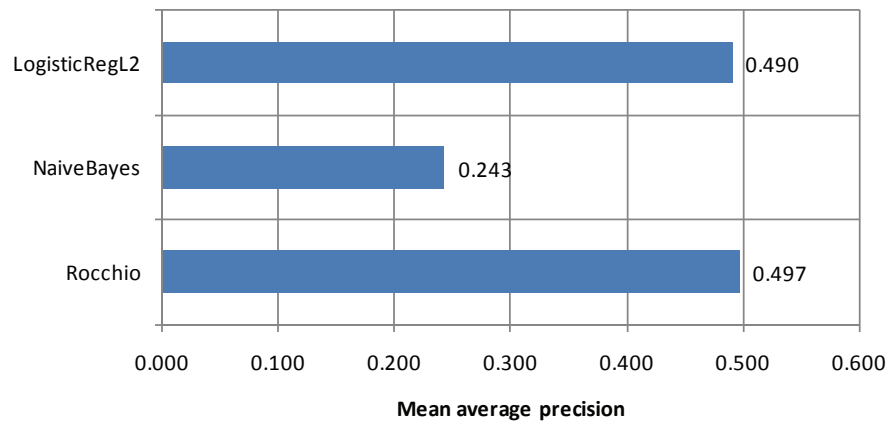


Figure 5.2. Reuters-21578 retrieval MAP evaluation.

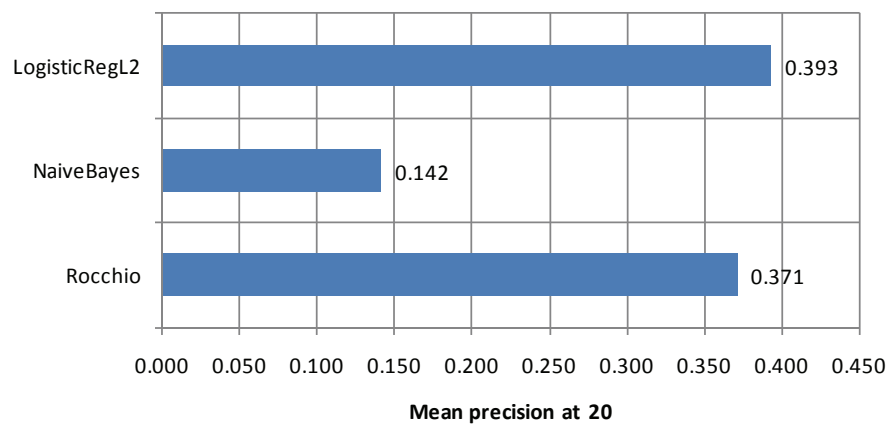


Figure 5.3. Reuters-21578 retrieval MP@20 evaluation.

However, $MP@20$ values on Figure 5.3 show that logistic regression is actually more selective than Rocchio because it can do better on the top 20 retrieved documents: logistic regression obtained 39.3% while Rocchio obtained only 37.1%. Interpolated precision-recall curves, Figure 5.4, offer a more detailed comparison of the models and confirm that logistic regression and Rocchio are very similar.

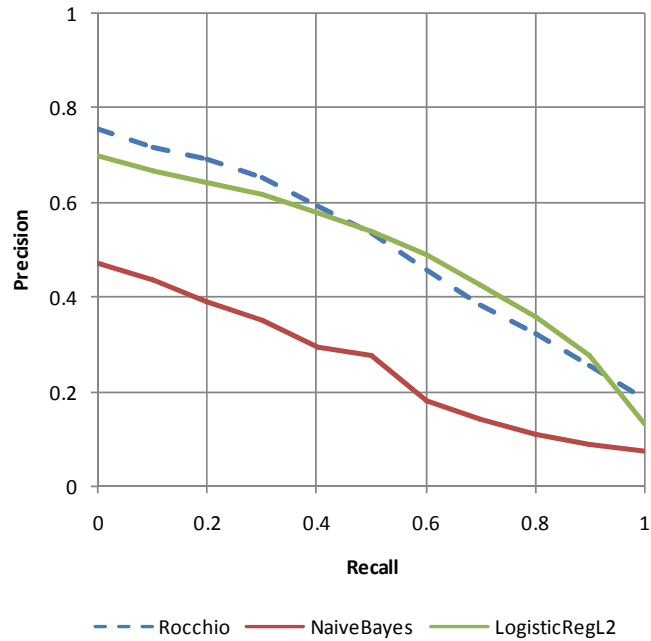


Figure 5.4. Interpolated precision-recall curve evaluation on the Reuters-21578.

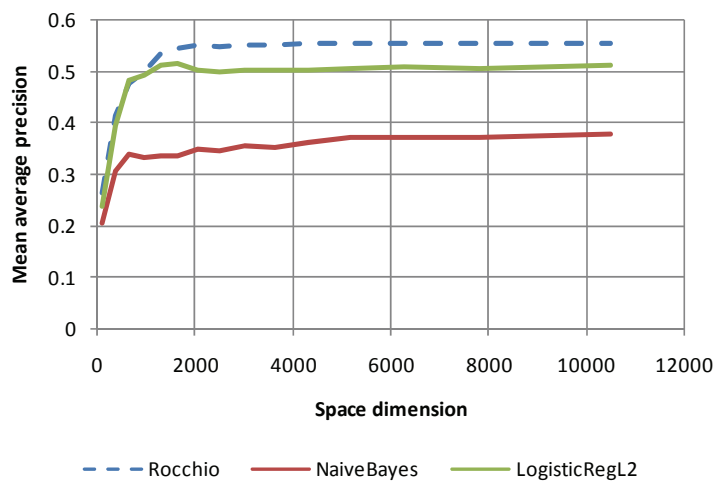


Figure 5.5. Retrieval precision for different space dimensions (text models).

Model Complexity Analysis

We also studied the effect of the optimal space dimensionality by measuring the MAP on different spaces. The different multi-modal feature spaces were obtained by progressively adding new terms according to the average mutual information criterion.

Figure 5.5 shows that after some number of terms (space dimension) precision do not increase because the information carried by the new terms is already present in the previous ones. The graph confirms that Rocchio is consistently better than logistic regression. Note that the MDL point (972 terms) achieves a good trade-off between the model complexity and the model retrieval effectiveness.

5.5.4 Image-Only Models

The image-only models experiment on the Corel Images collection evaluated the dense data processing part of the framework. The multi-modal feature space was created with the hierarchical EM algorithm described in Chapter 4. The different multi-modal feature spaces were obtained by concatenating different colour and texture representations. As before, we evaluated all linear models that we presented in this chapter.

Retrieval Effectiveness

We first applied the MDL criterion to select a multi-modal feature space and then ran the retrieval experiments for all linear models. The space selected by the MDL criterion has 2,989 dimensions.

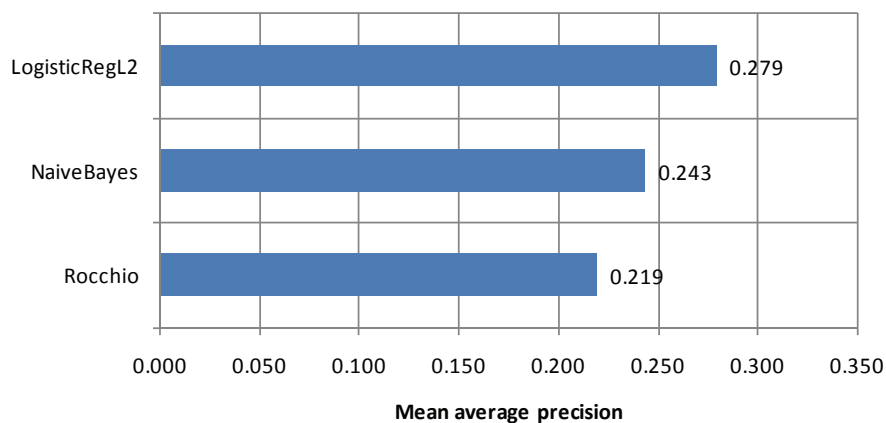


Figure 5.6. Corel retrieval MAP for different keyword models.

The MAP measures shown in Figure 5.6 shows that the best performance is achieved by the logistic regression models with a 27.9%, followed by naïve Bayes with 24.3% and Rocchio with 21.9%. The MP@20 measures in Figure 5.7 show that both naïve Bayes and logistic regression are

affected similarly. However, the Rocchio classifier is less selective as the decrease in retrieval accuracy shows (from 21.9% to 10.1%). Contrary to the Reuters collection, the more complex structure of Corel Images dataset has affected the performance of the Rocchio classifier. Thus, both naïve Bayes and, more specifically, logistic regression can better capture the structure of this data. The interpolated precision-recall curves in Figure 5.8 show that logistic regression is better than Rocchio and naïve Bayes across most of the recall area.

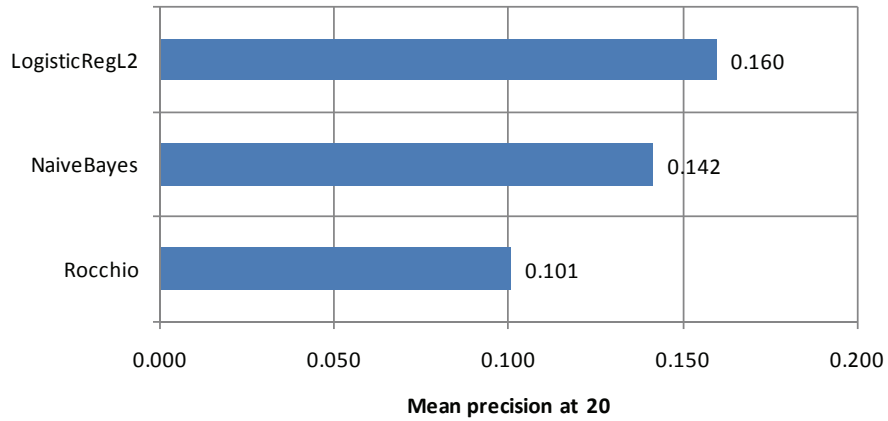


Figure 5.7. Corel retrieval MP@20 for different keyword models.

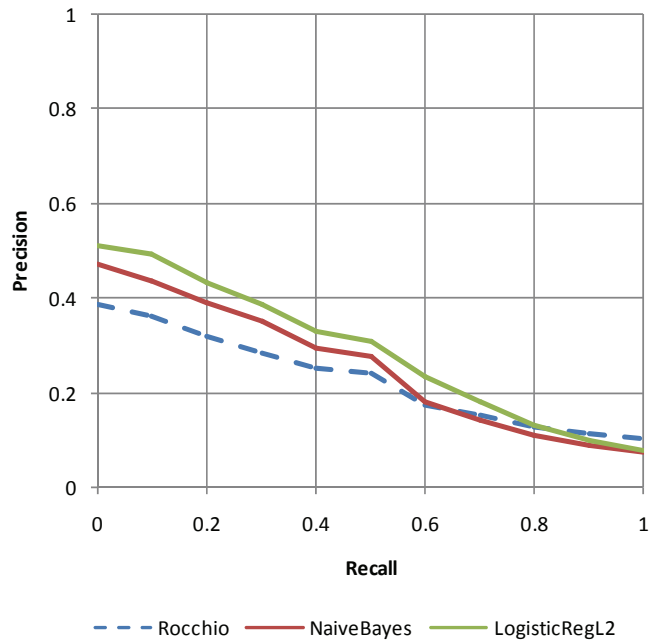


Figure 5.8. Interpolated precision-recall curves for different keyword models.

Results on this collection are more in agreement with what one would expect from the complexity of each model. Naïve Bayes applies a Gaussian on each dimension of the feature space, which reveals to be a more accurate assumption than the single cluster assumption made by the Rocchio classifier. Finally, logistic regression can better capture the non-Gaussian patterns of the data and achieve a better performance.

Algorithm	MAP	L
Cross-Media Relevance Model (Jeon, Lavrenko and Manmatha 2003)	16.9%	179
Continuous-space Relevance Model (Lavrenko, Manmatha and Jeon 2003)	23.5%	179
Naïve Bayes	24.3%	179
LogisticRegL2	27.9%	179
Non-parametric Density Distribution (Yavlinsky, Schofield and Rüger 2005)	28.9%	179
Multiple-Bernoulli Relevance Model (Feng, Lavrenko and Manmatha 2004)	30.0%	260
Mixture of Hierarchies (Carneiro and Vasconcelos 2005)	31.0%	260

Table 5.1. MAP comparison with other algorithms (Corel).

Table 5.1 compares some of the published algorithms' MAPs on the Corel collection. Note that some algorithms consider keywords with only training 1 example and 1 test example, thus resulting in 260 keywords instead of the 179 keywords. Methods that used the 260 keywords are some type of non-parametric density distributions that can easily model classes with a small number of examples. This table also shows how the proposed algorithm achieves a retrieval effectiveness that is in the same range as other state-of-the-art algorithms.

Model Complexity Analysis

Figure 5.9 depicts the evolution of the mean average precision with the dimensionality of the multi-modal feature space. Each point on the curve reflects the different levels of model complexities of the output of the hierarchical EM. Remember that the multi-modal feature space is the concatenation of the hierarchical EM Gaussian mixture models of the different feature subspaces. We concatenate sub-spaces with a similar number of level of complexity, e.g., GMMs with the same number of components per feature subspace.

For low dimensional multi-modal spaces the MAP for all models are quite low. Only when the dimensionality increases does the MAP achieve more stable values. The MAP stabilizes because the more complex GMMs models do not allow achieving a better discrimination between the relevant and non-relevant examples. The same phenomenon was observed on the Reuters collection.

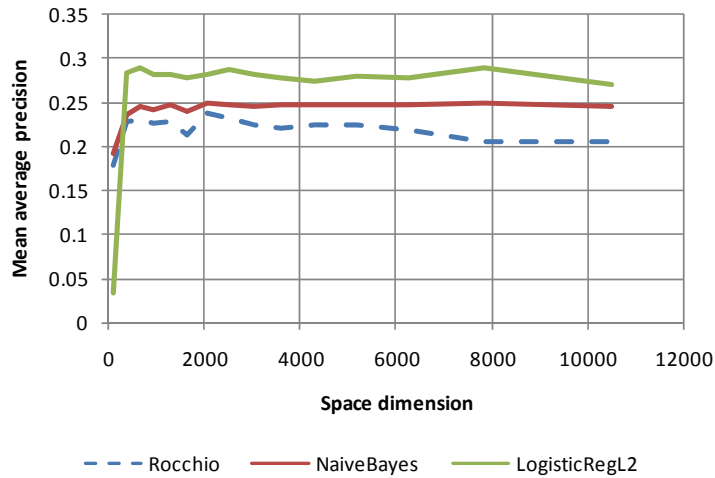


Figure 5.9. Retrieval precision for different space dimensions.

5.5.5 Multi-Modal Models

For the multi-modal models we proceeded in the same way as for the other single-medium experiments with the difference that we deployed single-media and multi-modal experiments to compare and analyse the information value of each modality.

Retrieval Effectiveness

We first applied the MDL criterion to select a multi-modal feature space and then ran the retrieval experiments for all linear models. The space selected by the MDL criterion has 5,670 dimensions for the visual modality, 10,576 for the text modality, and the multi-modal space has a total of 16,247 dimensions. For the text modality the MDL selects the maximum number of terms because some of the key-frames have no ASR.

Figure 5.10 and Figure 5.11 present a summary of the retrieval effectiveness evaluation in terms of MAP and MP@20, respectively. All types of keyword models show the same variation with respect to each modality: text based models are always much lower than the image based models, and the difference between image based models and multi-modal models is always small. Moreover, logistic regression models are always better than naïve Bayes and Rocchio. This confirms previous knowledge that TRECVID collection is more difficult and its data exhibit a more complex structure, which is why logistic regression can exploit the non-Gaussian patterns of data: it achieves 20.2% MAP on the text-only experiment, 27.3% on the image-only experiment and 29.5% on the multi-modal experiment.

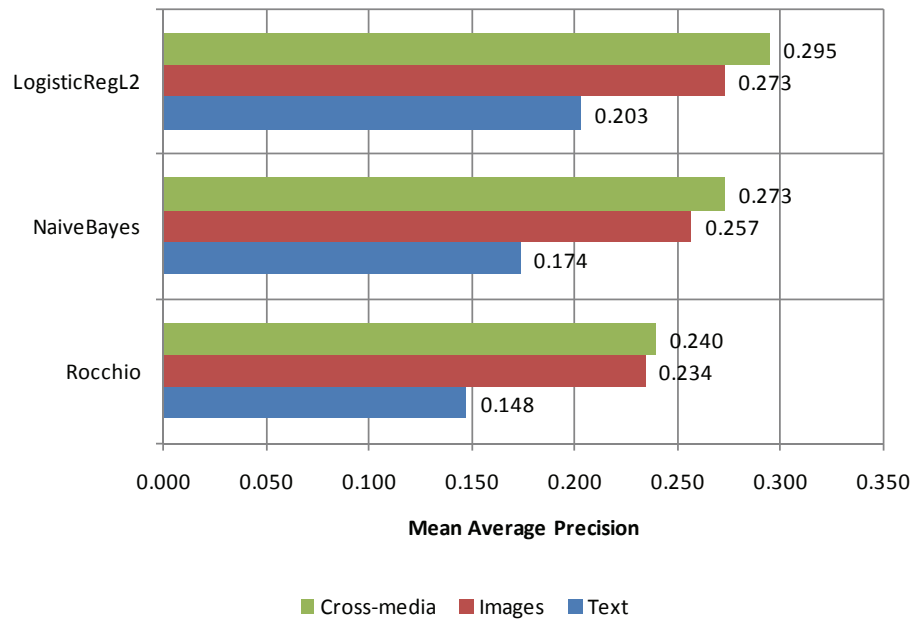


Figure 5.10. MAP by different modalities (TRECVID).

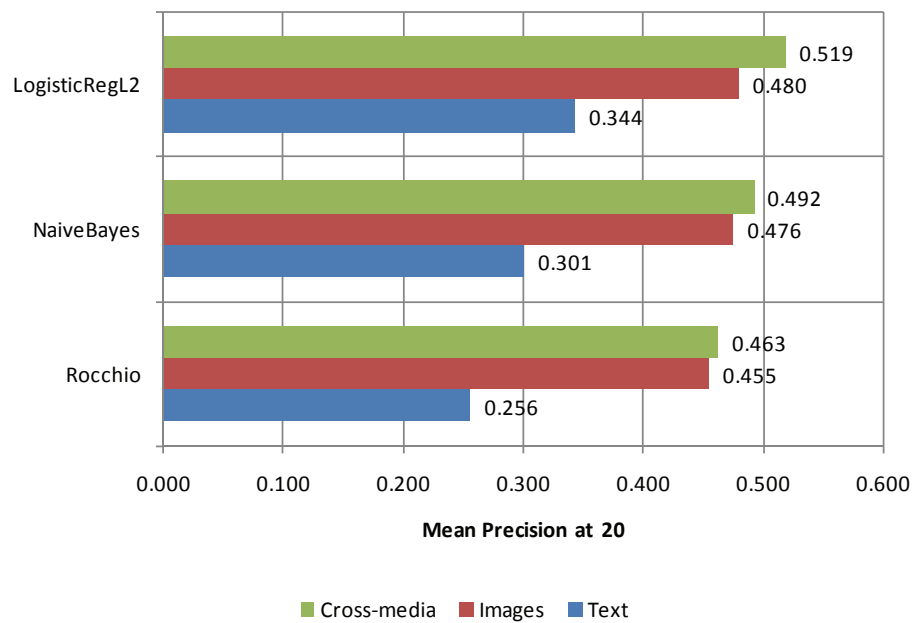


Figure 5.11. MP@20 by different modalities (TRECVID).

Text based models, Figure 5.12, exhibit a predictable behaviour: Rocchio is the less effective model, and logistic regression is the most effective model for all values of recall. However, for values of recall higher than 70%, all models are very similar. Image based models, Figure 5.13, present a similar behaviour but the difference between the Rocchio and the naïve Bayes model is very small. It is also possible to observe that there is a significant difference between these two models for values of recall between 10% and 90%. Multi-modal models, Figure 5.14, show that naïve Bayes models better exploit the higher number of information sources than the Rocchio classifier. This is not a surprise as naïve Bayes considers individual dimensions, and the data structure is more complex than the spherical structure assumed by Rocchio. Also related to this phenomenon is the retrieval effectiveness obtained by the logistic regression model.

Finally, Figure 5.15 compares the logistic regression model on the different modalities. The first phenomenon to note is the difference between the text modality and the images modality. We believe that text-only models achieved such a low performance because some of the documents do not contain any text, and most concepts are more directly related to visual features than to text features. Multi-modal models perform better than the best single-media based models, which was a predictable behaviour given the increase in the number of predictors. However, this difference is not as big as we expected initially. We believe that the larger number of predictors would require a more exhaustive cross-validation procedure.

Algorithm	MAP	Keywords	Modalities	Videos
LogisticRegL2	27.3%	39	V	English
Non-parametric Density Distribution (Yavlinsky, Schofield and Rüger 2005)	21.8%	10	V	All
LogisticRegL2	29.5%	39	V+T	English
SVM (Chang et al. 2005)	26.6	10	V+T	All

Table 5.2. MAP comparison with other algorithms (TRECVID).

Table 5.2 compares the proposed algorithm to two TRECVID submissions that attained an MAP above the median and all keywords are modelled with the same algorithm (some TRECVID systems employ a different algorithm for each keyword). Note that our results were obtained for more keywords (39 instead of 10) and less training data (just English), so, results are a rough indication of how our method compares to others. We limited the amount of training data due to computational reasons. However, as we can see from the table, the proposed approach is competitive with approaches that were trained in more advantageous conditions (fewer keywords).

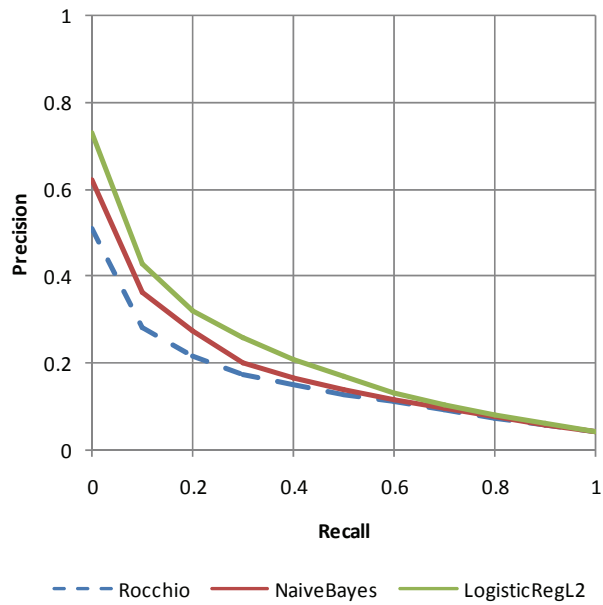


Figure 5.12. Interpolated precision-recall curve for the text models (TRECVID).

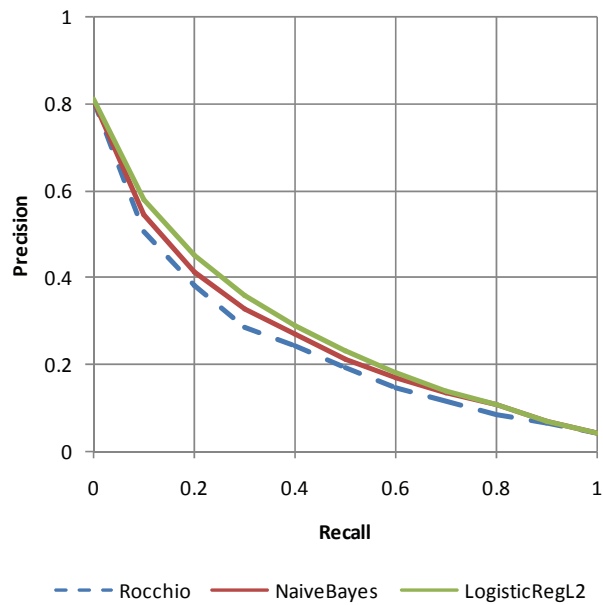


Figure 5.13: Interpolated precision-recall curve for image models (TRECVID).

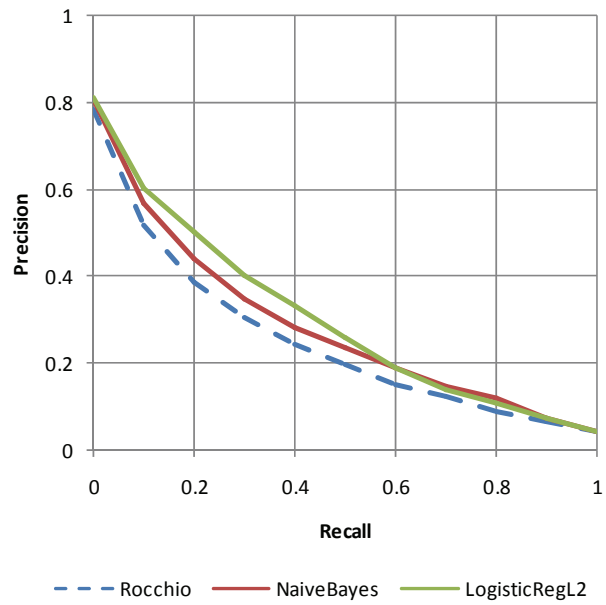


Figure 5.14. Interpolated precision-recall curve for multi-modal models (TRECVID).

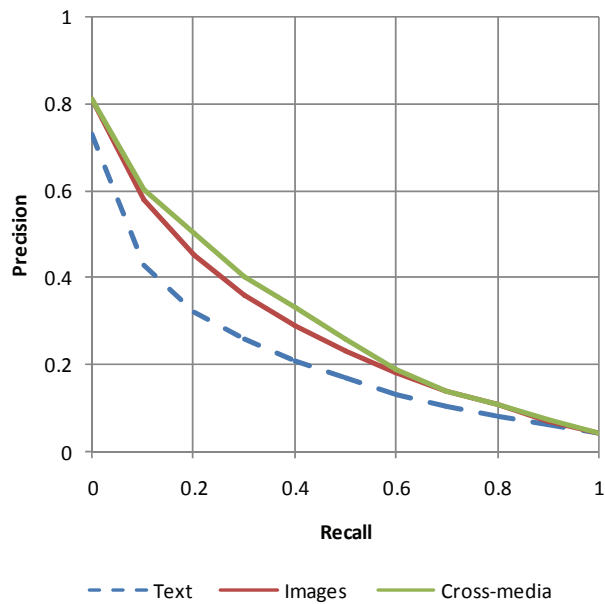


Figure 5.15. Interpolated precision-recall curves for different modalities (TRECVID, LogisticRegL2).

Model Complexity Analysis

For the second experiment we studied the effect of the complexity of the feature space transformations – the number of dimensions of the optimal feature space. Figure 5.16 illustrates the text-based models' retrieval effectiveness as new terms are added to the optimal feature space. The order by which terms are added is determined by the average mutual information. Retrieval effectiveness improves constantly but at a slower rate and with a different trend than for the Reuters collection. Again, we believe that this is related to the fact that some documents have no text and that TRECVID data is more complex.

Image based models, Figure 5.17, show an identical trend to the Corel collection. For a small number of dimensions the retrieval effectiveness is quite low and it quickly increases until a given dimensionality. The MAP achieves a stable range of values after around 5,000 dimensions and is not affected by the addition of new dimensions to the feature space.

Multi-modal based models, Figure 5.18, exhibit a more irregular trend than the single-media models. The higher dimensionality and features heterogeneity might be the cause for this phenomenon. The differences between the three models is related to the respective modelling capabilities: Rocchio assumes a spherical structure which reveals to be too simplistic for this data; naïve Bayes assumed independent dimensions, which is also not the best model for this data; finally, logistic regression further exploits feature dimensions interactions with linear combinations of them. Logistic regression, with an adequate cross-validation procedure, revealed to achieve the best retrieval effectiveness.

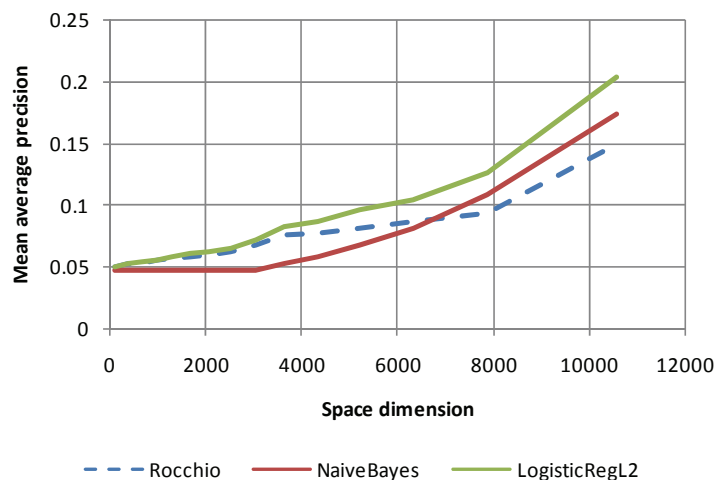


Figure 5.16. Retrieval precision for different space dimensions (TRECVID, text).

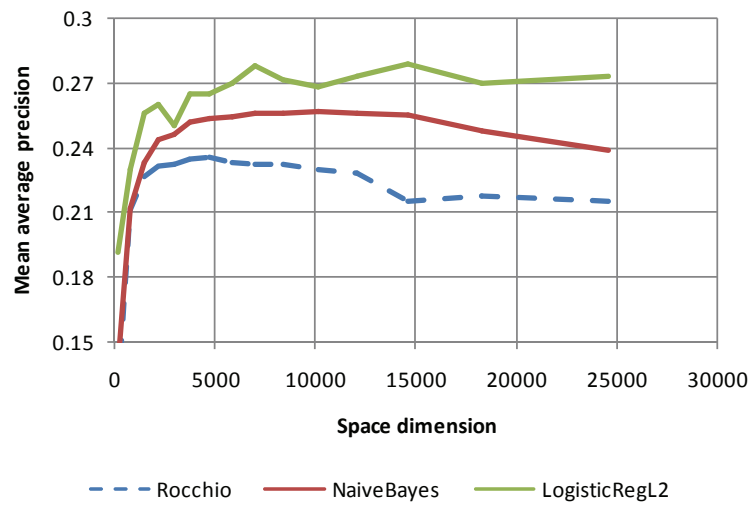


Figure 5.17. Retrieval precision for different space dimensions (TRECVID, images).

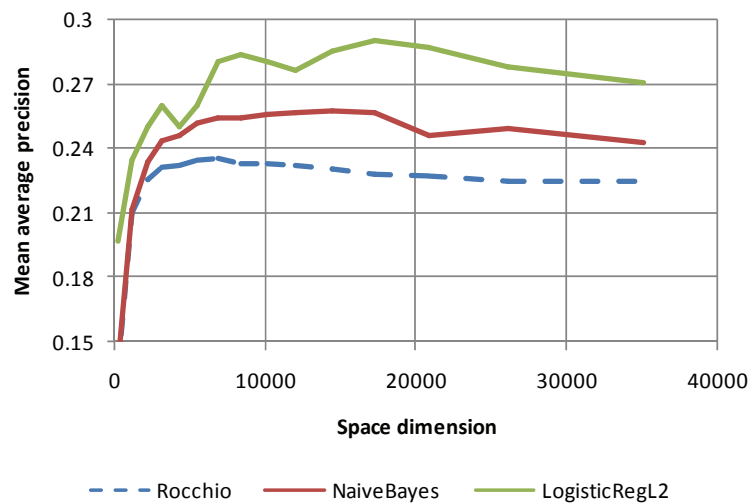


Figure 5.18. Retrieval precision for different space dimensions (TRECVID, multi-modal).

5.6 Conclusions and Future Work

The creation of the multi-modal feature space is a generalization procedure which results in a trade-off between accuracy and computational complexity. Thus, the described algorithm offers an appealing solution for applications that require an information extraction algorithm with good precision, scalability, flexibility and robustness.

The novelty of the proposed framework resides in the simplicity of the linear combination of the heterogeneous sources of information that were selected by the minimum description length criterion.

5.6.1 Retrieval Effectiveness

The performed experiments show that our framework offers a performance in the same range as other state-of-the-art algorithms. It was not surprising to see that logistic regression attains better results than naïve Bayes at the expense of a higher learning cost. Table 5.1 summarized the performance of several alternative algorithms on the Corel dataset with just slight changes (the number of keywords and the type of features). Text and image results are quite good while multimodal experiments were affected by the noise present on the speech text and by the higher number of parameters to estimate.

Results on the TRECVID collection are more difficult to compare because participants apply different important changes that obfuscate algorithms comparison: correction of ground-truth provided by NIST; use of different low-level features; different keywords are modelled with different algorithms, e.g., *sky* might be modelled with a GMM, *face* with an SVM, and *vegetation* with a k-NN. Despite this fact we presented a summary in Table 5.2 that shows how our method attains a retrieval effectiveness that is in the same range as other state-of-the-art methods.

5.6.2 Model Selection

The algorithm's immunity to over-fitting is illustrated by the MAP curve stability as the model complexity increases. Logistic regression can be interpreted as ensemble methods (additive models) if we consider each dimension as a weak learner and the final model as a linear combination of those weak learners. This means that our model has some of the characteristics of additive models, namely the observed immunity to overfitting. It is interesting to note that the simple naïve Bayes model appears to be more immune to overfitting than the logistic regression model. This occurs because the optimization procedure fits the model tightly to the training data favouring large regression coefficients, while the naïve Bayes avoids overfitting by computing the weighted average of all codewords (dimensions). Note that when fitting the model we are minimizing a measure of the model log-likelihood (the average classification residual error) and not a measure of how

documents are ranked in a list (average precision). The mean average precision is the mean of the accumulated precision over a ranked list. Thus, we believe that if we trained our models with average precision as our goal metric, the retrieval results on the test set would improve.

5.6.3 Computational Scalability

Since the optimal feature space is common to all keywords the transformation must be computed only once for all keywords. Thus, the resources required to evaluate the relevancy of a multimedia document for each keyword are relatively small. During classification, both time and space complexity of the data representation algorithms is given by the number of Gaussians (clusters) selected by the model selection criteria. The computational complexity of linear models during the classification phase is negligible, resulting in a very low computational complexity for annotating multimedia content and making it quickly searchable.

The computational complexity during the learning phase is dominated by the hierarchical EM algorithm of mixture of Gaussians and the cross-validation method. The worst-case space complexity during learning is proportional to the maximum number of clusters, the number of samples, the dimension of each feature, and the total number of cross-validation iterations and folds. I consider this cost to be less important because the learning can be done offline.

Apart from the mixture of hierarchies (Carneiro and Vasconcelos 2005) all other methods are some sort of kernel density distributions. It is well known (Hastie, Tibshirani and Friedman 2001) that the nature of these methods makes the task of running these models on new data computationally demanding: the model corresponds to the entire training set meaning that the demand on CPU time and memory increases with the training data.

For these reasons, our approach has a lower computational complexity during the classification phase. It has a bearing on the design of image search engines, where scalability and response time is as much of a factor as the actual mean average precision of the returned results: Table 7.4 in Chapter 7 illustrates how the low computational complexity enables a new search paradigm that requires the detection of multiple concepts on-the-fly.

5.6.4 Semantic Scalability

Assuming that the used set of keywords is a faithful sample of a larger keyword vocabulary it is expected that one can use the same optimal feature space to learn the linear model of new keywords and preserve the same models. Note that the optimal feature space is a representation of the data feature space: it is selected based on the entire data and independently of the number keywords. The consequence of this design is that systems can be semantically scalable in the sense that new keywords can be added to the system without affecting previous annotations.

5.6.5 Future Work

The evaluation of the presented linear models has uncovered new issues that can be tackled by further researching the following topics:

- **L₁ regularized logistic regression:** sparse models are known to perform better than the smoothed version of logistic regression, e.g., relevance vector machines or support vector machines. This would allow us to use arbitrary dimensions of the feature space and discard the ones that are not in use thus reducing the computational complexity.
- **Replace cross-validation:** cross-validation based model selection is computationally very complex and demands large computational resources. Other methods for linear models exist that can reduce the model selection cost such as the newly proposed method to follow regularization paths (Park and Hastie 2007).
- **Use other features** (SIFT, text relations, etc): we limited the set of features to very simple ones as our focus was on the models and not on the features. However, it would be interesting to evaluate the usefulness of more semantic features such as WordNet or other visual grammars.

Part 2
Searching Semantic-Multimedia

Searching Multimedia

6.1 Introduction

In the classic information retrieval search paradigm the user transforms some information need into a system query, and the system replies with the required information. Unlike text documents, multimedia documents do not explicitly contain symbols that could be used to express an information need. This problem has roots in two different aspects:

- **Richness of multimedia information:** visual and audio information can communicate a wide variety of messages, feelings and emotions; temporal and spatial structure adds organization and usability.
- **Expressiveness of the user query:** systems have always forced humans to describe their information need in some query language. However, not all information needs are easily expressed.

Multimedia information retrieval systems are best at processing user queries represented by mathematical expressions, and not everyone have the same skills at expressing ideas, emotions and feelings in such a formal way. While in text retrieval we express our query in the format of the document (text), in multimedia retrieval systems this is more difficult due to semantic ambiguities. The user is not aware of the low-level representation of multimedia, e.g., colour, texture, shape features, pitch, volume or tones. Instead the user is often more interested in the semantic richness of multimedia information. This demands a search system that relies on a high-level concept representation of multimedia, thus, providing a semantic layer to multimedia documents. Figure 6.1 illustrates how an image is represented by both low-level features and high-level features (keyword

annotations and metadata).

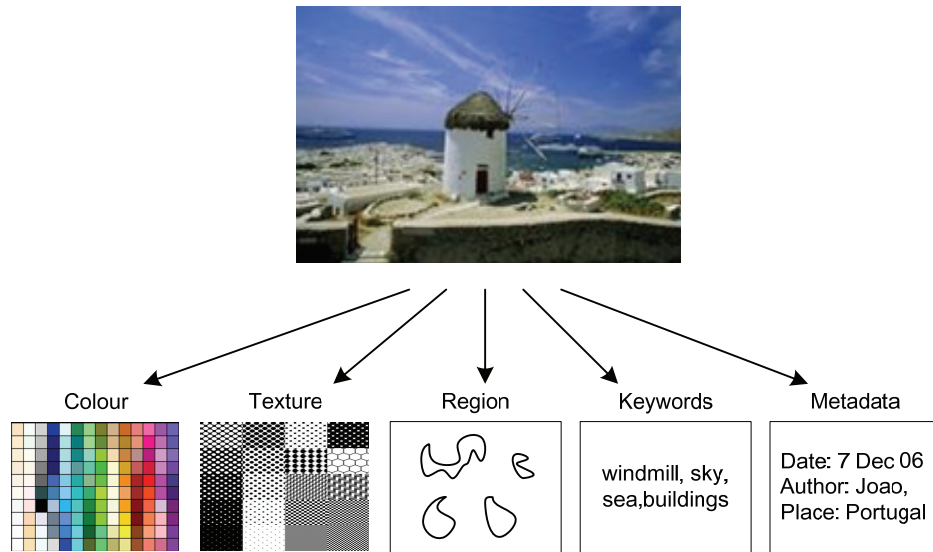


Figure 6.1. Examples of search spaces of visual information.

The goal now is to explore new ways of applying this semantic layer to improve search. The semantic layer is created with the output of the keyword models proposed in the first part of this thesis. It creates a keyword space that organizes multimedia according to their semantics. Thus, it allows users to search by keyword and by semantic example. In this chapter and the following I will address the problem of search by example in keyword spaces, i.e., search by semantic example.

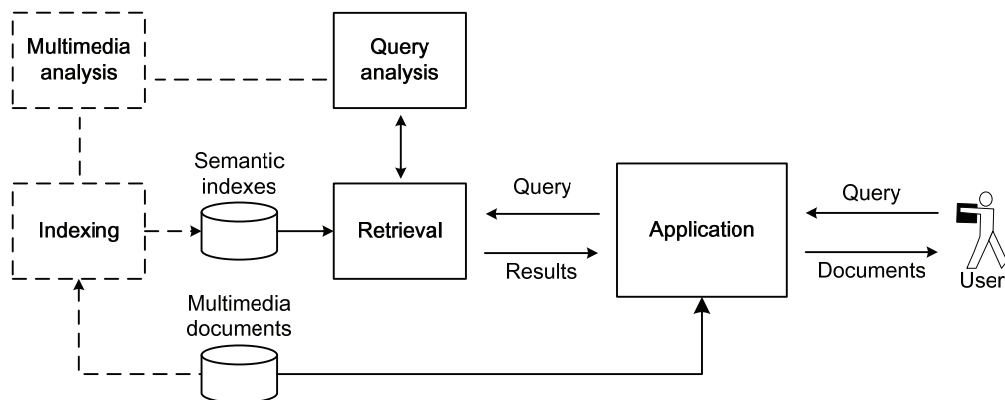


Figure 6.2. The scope of semantic query-processing.

The architecture of a multimedia information retrieval system defined in Chapter 1 is reproduced in Figure 6.2 where the scope of Chapters 6 and 7 is highlighted (solid lines). In this chapter I will review these techniques and in the following chapter I will present a framework for searching semantic-multimedia spaces.

6.2 Content based Queries

Early research in multimedia retrieval produced several systems that allowed users to search multimedia information by its content. The user would provide an example image (or an audio file) or a sketch image (or a melody humming) containing what they wanted to search for. QBIC (Flickner et al. 1995) is by far the best known of such systems but several other systems appeared at the same time: VisualSeek (Smith and Chang 1996); Informedia (Wactlar et al. 1996); PicHunter (Cox et al. 1996); Virage (Bach et al. 1996); MARS (Ortega et al. 1997); SIMPlicity (Wang, Li and Wiederhold 2001). This multitude of systems explored new techniques and introduced others into the area of multimedia retrieval. Many of these are present in systems produced nowadays. For example, VisualSeek was one of the pioneers in Web image crawling and search, and MARS introduced a new relevance feedback method that became highly popular (Rui et al. 1998).

All these systems implement a content based search paradigm where query processing methods are based on the principle that information needs can be expressed by example images or sketch images provided by the user. This is a good starting point and if users are able to provide examples then it would be much easier for the system to find relevant documents.

Query processing algorithms start by analysing the provided examples and extract low-level features from them. Once user examples are represented by low-level features (colour, texture, regions, motion, pitch, tones or volume features), the next step is to rank the database documents by similarity. In this process there are two aspects that are fundamental to query processing in content based search. The first one is the reduction of a user example to a set of low-level features. This implies that the user understanding of the provided example is captured by the extracted low-level features. The second aspect is the subjective notion of similarity. There is always some ambiguity as to what exactly the provided example illustrates. This problem of visual similarity was studied by Ortega et al. (1997) and in many other cases, e.g., (Swain and Ballard 1991; Heesch and Ruger 2004; Vasconcelos 2004).

Low-level features capture part of the knowledge represented in a multimedia document, and there are situations where search by colour, texture or shape is an excellent solution. However, low-level features might not be the ideal representation when the search is semantic and the goal is to find examples of cars, dogs, etc. This is the so called semantic gap. To overcome this problem two types of methods have been proposed: semi-automatic methods that rely on user feedback guiding the system with positive and negative examples (relevance feedback), and automatic methods that rely on high-level feature representations of information (semantic based queries).

6.3 Relevance Feedback

Relevance feedback systems (Rocchio 1971) allow the user to compose a set of visual positive examples that are different representations of the same semantic request. Relevance feedback tries to iteratively specify the semantic characteristics of the intended results by adding semantically relevant examples and removing semantically non-relevant examples from the working model. Positive and negative examples are obtained from user feedback in different ways:

- **Explicit feedback** is obtained by having the user marking specific documents as relevant or non-relevant. This information allows the system to create a relevance model for each specific query; the MARS system proposed a popular relevance feedback technique (Rui et al. 1998).
- **Implicit feedback** is inferred from user interactions, such as noting which documents users select for viewing, and how long they view those documents. This approach is also known as long-term models because query logs are used to refine relevance models, see (Vasconcelos 2000).
- **Blind relevance feedback** is obtained by assuming that the top n documents in the result set are actually relevant; a query with those top examples is automatically resubmitted as positive examples.

Explicit relevance feedback is by far the most researched approach, differing mainly in the multimedia representation method that tries to mimic human perception. Yang et al. (2005) implemented a relevance feedback algorithm that works on a semantic space created from image clusters that are labelled with the most frequent concept in that cluster. Semantic similarity is then computed between the examples and the image clusters. Lu et al. (2000) proposed a relevance feedback system that labels images with the previously described heuristic and updates these semantic relations according to the user feedback. The semantic links between the examples and the keywords are heuristically updated or removed. Zhang and Chen (2002) followed an active learning approach, and He et al. (2003) applied spectral methods to learn the semantic space from user feedback.

Smeulders et al. (2000) summarized the research area of content based search and relevance feedback in their classic paper. Note the difference between content based queries where multimedia semantics is automatically represented as low-level features, and relevance feedback, where the user is inserted in the loop to better define multimedia semantics in terms of low-level features. The use of relevance feedback per se does not make the system aware of any semantics as

it still represents images by their low-level features. In my opinion content based queries are limited in the way information is represented: low-level features are not sufficient to represent the entire universe of interpretations that a user might have regarding a multimedia document. Instead users might be interested in searching multimedia by its semantic content.

6.4 Semantic based Queries

Systems that are aware of multimedia semantics have already flourished in the multimedia information retrieval community allowing different search paradigms. Figure 6.3 illustrates three different semantic search paradigms that users can exploit to satisfy their information need. These search paradigms work on a high-level feature space that is obtained through different methods. The semantic space is obtained either through some manual method, automatic method or semi-automatic method, e.g., relevance feedback.

Automatic algorithms are attractive as they involve a low analysis cost when compared to manual alternatives. Automatic methods are based on heuristics or on some pattern recognition algorithm. Heuristic techniques rely on metadata attached to the multimedia: for example, Lu et al (2000) analyse HTML text surrounding an image and assign the most relevant keywords to an image. Pattern recognition algorithms exploit low-level features extracted from the multimedia itself and create a model for each keyword that needs to be detected. Several techniques have been proposed in the literature: Feng, Lavrenko and Manmatha (2004) proposed a Bernoulli model with a vocabulary of visual terms for each keyword, Carneiro and Vasconcelos (2005) a semi-parametric density estimation based on DCT features of images, Magalhães and R uger (2007b) developed a maximum entropy framework to detect multi-modal concepts, while Snoek et al. (2006) proposed an SVM based multi-modal feature fusion framework. Chapter 3 discusses these methods in detail.

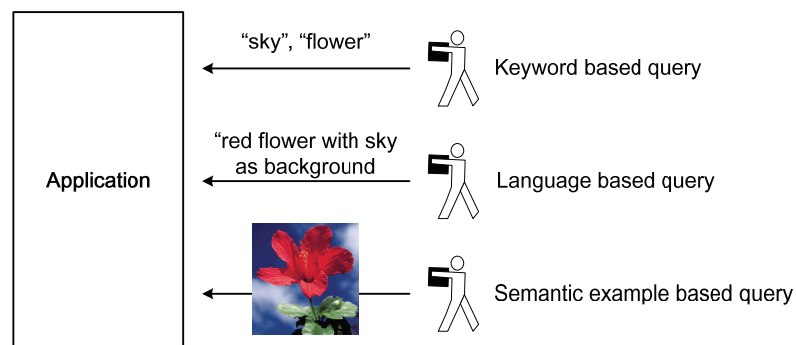


Figure 6.3. Semantic based search.

Thus, it is in this context that Chapter 6 and Chapter 7 study keyword spaces (created with the output of keyword detector algorithms) for multimedia retrieval by example.

6.4.1 Keyword based Queries

The direct application of keyword annotations, i.e. high-level features, allows the user to specify a set of keywords to search for multimedia content containing these concepts. This is already a large step towards more semantic search engines. Although quite useful in some cases this still might be too limiting: semantic multimedia content captures knowledge which goes beyond the simple listing of keywords. The interaction between concepts, the semantic structure and the context are aspects that humans rely on to express some information need. Natural language based queries and semantic example based queries explore these aspects.

6.4.2 Natural Language based Queries

In text IR systems the user can create text based queries by combining keywords with simple Boolean expressions as in inference networks (Turtle and Croft 1991) or by writing a natural language query expression (Croft, Turtle and Lewis 1991). These types of query expressions are now possible in multimedia information retrieval owing to algorithms that can detect multimedia concepts. Recently, Town and Sinclair (2004) proposed an ontology based search paradigm for visual information that allows the user to express his query as a sentence, e.g., “red flower with sky background”. It relied not only on the detection of concepts but also on the information stored in the ontology regarding concept and concept-relations.

6.4.3 Semantic Example based Queries

These type of approaches can produce good results but it puts an extra burden on users who now have to describe their idea in terms of all possible instances and variations, or express it textually. This requires creativity or expressiveness, which may be a limiting factor. Thus, in these cases users should be able to formulate a query with a semantic example of what they want to retrieve. Of course, the example is not semantic per se but the system will look at its semantic content and not only at its low-level characteristics, e.g., colour or texture. This means that the system will infer the semantics of the query example and use it to search the image database. Both database and query are analysed with the same concept extraction algorithm. Moving away from implementing query by semantic example as relevance feedback, Rasiwasia et al. proposed a framework to compute semantic similarity to rank images according to the current state of the query, (Rasiwasia, Vasconcelos and Moreno 2006; Rasiwasia, Moreno and Vasconcelos 2007). They start by extracting semantics with an algorithm based on a hierarchy of mixtures (Carneiro and Vasconcelos 2005) and compute the semantic similarity as the Kullback-Leibler divergence. Tesic et al. (2007) address the same problem but replace the Kullback-Leibler divergence by an SVM. The SVM uses the provided examples as the positive examples, and negative examples are randomly

sampled from the database. A cluster model of the database is used to sample negative examples from clusters where the positive examples have low probability. Their results show good improvements over text-only search. Following these steps, Natsev et al. (2007) explored the idea of using concept-based query expansion to re-rank multimedia documents. They discuss several types of methods to expand the query with visual concepts. Another approach to query expansion in multimedia retrieval by Haubold et al. (2006) uses lexical expansions of the queries.

Hauptman et al. (2007) present an estimation of the number of concepts that is required to fill the semantic gap. They use a topic search experiment to assess the number of required concepts to achieve a high precision retrieval system – their study suggests 3,000 concepts. This approach associates the success of semantic-multimedia IR to a single factor (number of concepts) and leaves several aspects of the problem, e.g., similarity functions and different querying paradigms, out of the analysis.

The described family of techniques allow ranking algorithms to work at a semantic level by extracting the concepts from both multimedia and users' queries examples. The second step in the ranking problem is to explore the semantic similarity between the users' examples and the multimedia documents. Thus, semantic similarity, computed either in a low-level feature space or a high-level feature space, is a corner stone in the ranking process.

6.4.4 Semantic Similarity

Semantic similarity tries to measure the difference in the meaning of the information of two documents. Two different approaches are popular: a distance function in a semantic space and a walk function in an ontology graph. Both methods can either use a predefined metric or can learn a metric based on some training data, e.g., (Yu et al. 2008). In the next chapter we will thoroughly discuss and analyse a set of predefined metrics in a keyword space. The second type of methods is based in an ontology that mirrors human knowledge. Smeaton and Quigley (1996) explored semantic distances between words for query expansion. They show that semantic distances based on WordNet offers a substantial improvement over traditional IR techniques. Benitez and Chang (2000) also explored ontology based methods to compute the semantic similarity between images.

6.5 Summary

In this chapter I motivated the application of keyword spaces to the problem of search by example and compared it to previous research. Several search paradigms delivering different ways of user information need expressiveness were discussed: content based queries (low-level features examples); relevance feedback (interactive); semantic based queries (keywords, natural language and

semantic examples). Early systems allowed the user to submit queries only based on low-level features, while most recent systems already allow the use of automatically extracted high-level features. I have emphasised semantic multimedia example based queries as this is one of the less studied methods and it is a natural application of the multimedia analysis algorithms described in the previous chapters.

Keyword Spaces

7.1 Keywords and Categories

Multimedia semantics is related to the way humans think and perceive multimedia information. The link between low-level features and high-level features is a known problem that has been addressed by a large body of work and is pointed as one of the main bottlenecks in semantic-multimedia information retrieval. In this chapter I address the problem of ranking multimedia by semantic similarity. This search paradigm allows the user to submit a single example image of a yellow flower and retrieve images of flowers of all colours, textures and backgrounds. This is possible because the search space does not represent multimedia by their low-level features but by their high-level concepts, e.g., flowers, mountains, river, or sky. It is in this context that I designed a search framework to study similarity ranking for search by semantic-multimedia example.

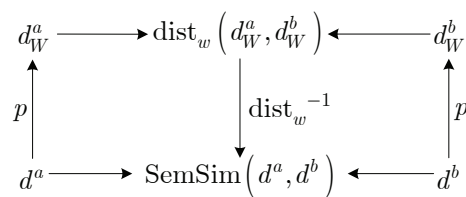


Figure 7.1. Commutative diagram of the computation of semantic similarity between two multimedia documents.

This scenario calls for a feature space capable of representing multimedia by its semantic content where semantic similarity is easily computed. Figure 7.1 depicts the process of computing the semantic similarity $\text{SemSim}(d^a, d^b)$ between multimedia a documents d^a and d^b . A multimedia document d^a is transformed into the keyword space by the $p : d^a \rightarrow d_W^a$ transformation. In this keyword space, a multimedia document d^a is represented by the vector d_W^a .

containing keyword scores. These scores indicate the confidence that a keyword is present in the document. Now, in this keyword space the distance $\text{dist}_w(d_W^a, d_W^b)$ between vectors d_W^a and d_W^b is inversely proportional to the semantic dissimilarity⁶ between documents d^a and d^b , i.e., $1 / \text{SemSim}(d^a, d^b)$. In this chapter we study the following aspects of this process:

- Manual versus automatic methods of transforming a multimedia document into the keyword space, i.e., the $p : d^a \rightarrow d_W^a$ transformation.
- Functions to compute the semantic dissimilarity as the distance $\text{dist}_w(d_W^a, d_W^b)$ between two keyword vectors.
- The influence of the keyword space dimensionality on the distance functions $\text{dist}_w(d_W^a, d_W^b)$.
- The influence of manual annotations accuracy on the computation of semantic similarity functions.

It is in this context that we designed a framework to search multimedia by semantic similarity. As mentioned before, the keyword vectors can be obtained by manual or automatic methods, which we define formally as:

- **User keywords:** a user manually annotates multimedia with keywords representing meaningful concepts present in that multimedia content.
- **Automatic keywords:** an algorithm infers multimedia keywords and a corresponding confidence representing the probability that a given concept is present in that multimedia content.

Figure 7.2 illustrates some of the images on the Flickr web site annotated by a user with the keyword “*London*”. These images can be further grouped into themes concerning the same idea: (1) *London touristic attractions*; (2) *London’s river Thames*; (3) *London metro*; (4) *London modern art*. Each one of these themes is a row of images in Figure 7.2. Formally we define categories as:

- **Categories** are groups of multimedia documents whose content concern a common meaningful theme, i.e., documents in the same category are semantically similar.

The above definitions create two types of content annotations – at the document level

⁶ Distance is equivalent to the inverse of similarity: large distances imply low similarity and small distances imply high similarity.

(keywords) and at the group of documents level (categories). Because both keywords and categories describe the content of multimedia one would assume that categories can be inferred from keywords. For example, given a query image depicting the *Big Ben* the system would retrieve other images belonging to the same category, “*London touristic attractions*”, and not necessarily visually similar.

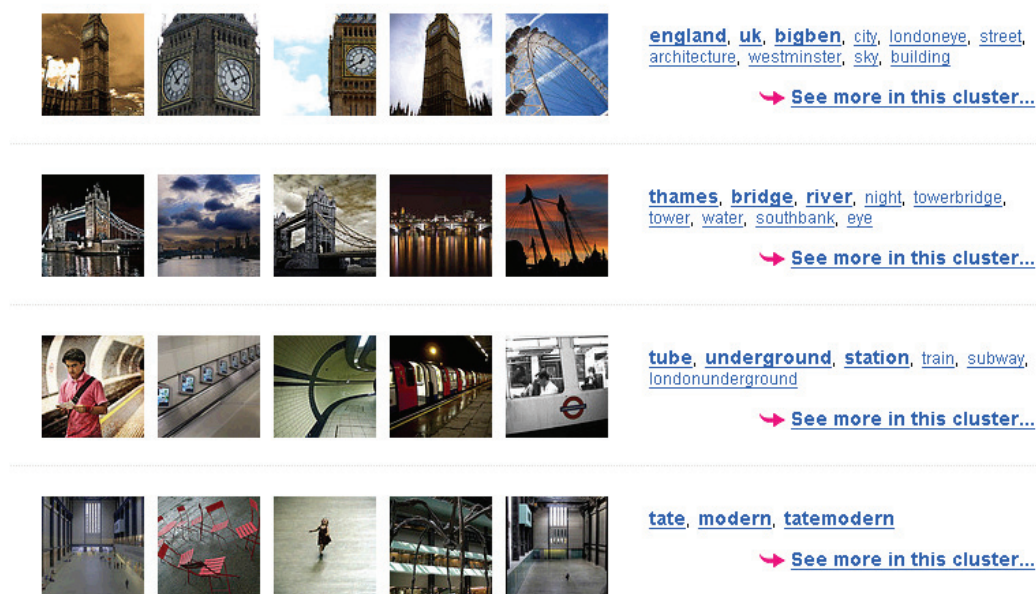


Figure 7.2. Example of Flickr images annotated with the keyword London.

In our experimental framework, keywords and categories of multimedia documents are defined by each collection ground truth: keywords are used to compute semantic similarity and categories are used to evaluate semantic similarity.

Next I formalize the idea of keyword spaces, followed by the implementation description of our semantic-multimedia search system. Section 7.3 describes how keyword vectors are computed with a naïve Bayes algorithm (automatic keywords) or are obtained from the ground truth labels of the collection (user keywords). We then apply noise to the user keywords to study the influence of different levels accuracy. Once documents are represented in the keyword space the user can select or submit a query document (Section 7.4). A semantic similarity function is used to find documents from the same unknown category (Section 7.5). Section 7.6 presents the evaluation experiments of the keyword space. Experiments were done on Corel Images and TRECVID data.

7.2 Defining a Keyword Space

Our goal is to devise a search space capable of representing documents according to their

semantics and with a defined set of semantic operations. Semantic spaces are similar to other feature spaces like colour or texture feature spaces where the space structure replicates a human notion of colour or texture similarity (assuming image documents). The distinction is clear: while in the first case images are organized by their texture or colour similarity, in semantic spaces images are organized by their semantic similarity. Figure 7.3 illustrates a visual semantic space where each dimension corresponds to a given keyword and images that are semantically similar are placed in the same neighbourhood. The usefulness of such a semantic space ranges from search-by-example to tag-suggestion systems and recommender-systems.

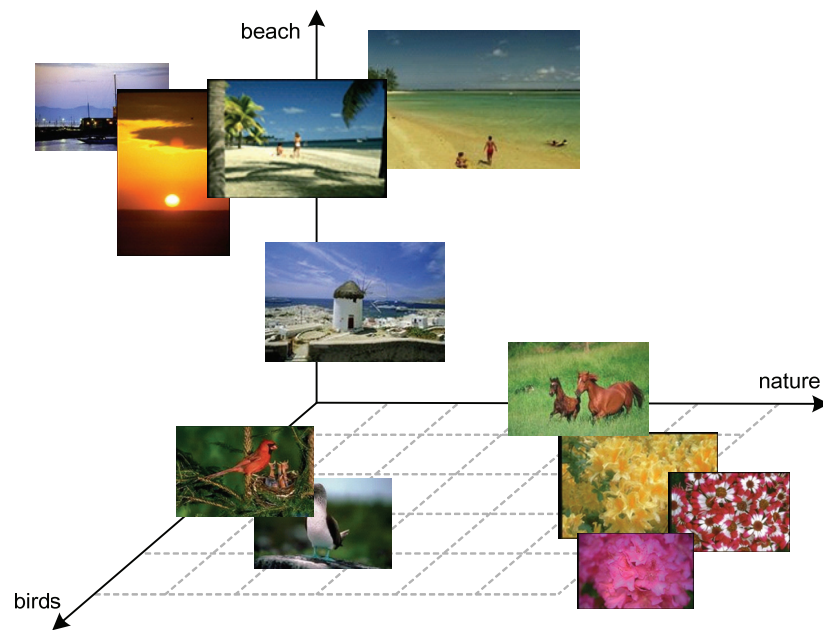


Figure 7.3. A keyword space with some example images.

In this setting, and reusing the notation defined in Chapter 4, we represent a multimedia document as

$$d = (d_T, d_V, d_W) = (d_f, d_W), \quad (7.1)$$

where d_W corresponds to the document keyword annotations and d_f to the document low-level features $d_f = (d_T, d_V)$. These two representations form two distinct feature spaces, e.g., in the first case an image is represented by its texture or colour features, in the second case the same image is represented by its semantics in terms of keywords. A keyword space for searching multimedia by semantic similarity is defined by the following properties:

- **Vocabulary:** defines a lexicon

$$\mathcal{W} = \{w_1, \dots, w_L\} \quad (7.2)$$

of L keywords used to annotate multimedia documents.

- **Multimedia keyword vectors:** a multimedia document d is represented by a vector

$$d_W = (d_{W,1}, \dots, d_{W,L}) \in [0,1]^L \quad (7.3)$$

of L keywords from the vocabulary \mathcal{W} , where each component d_i corresponds to the likelihood that keyword w_i is present in document d .

- **Keyword vectors computation:** the keyword vector can be computed automatically or provided by a user. Section 7.3 discusses and compares both methods.
- **Semantic dissimilarity:** given a keyword space defined by the vocabulary \mathcal{W} , we define semantic dissimilarity between two documents as

$$\text{dissim}_w : [0,1]^L \times [0,1]^L \rightarrow \mathbb{R}_0^+, \quad (7.4)$$

the function in the L dimensional space that returns the distance between two keyword vectors. Section 7.5 presents several distance functions.

Given the above definitions it is easy to see that for a query example $q = (q_f, q_W)$ and a candidate document $d = (d_f, d_W)$, the semantic similarity between documents is computed as the inverse of the dissimilarity $\text{dissim}_w(q_W, d_W)$ between the corresponding keyword vectors.

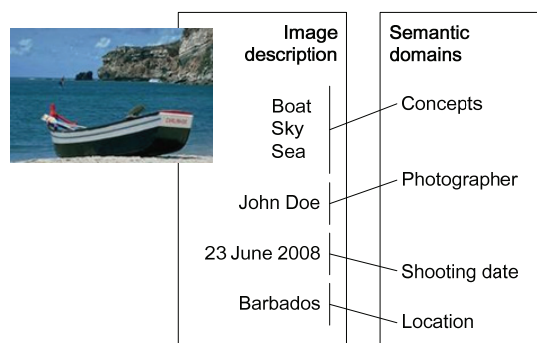


Figure 7.4. A multimedia document description.

The lexicon of keywords corresponds to dimensions of the keyword space, allowing documents to be represented with varying types of information according to the type of keyword, e.g., visual concepts, creation date and author. Figure 7.4 illustrates how documents can be described with

different representation schemes. This richness of expressivity might confuse ranking algorithms – the same document can have multiple interpretations each one giving more emphasis to different sets of keywords. Thus, by limiting the semantic representation to a subset of the document semantics one defines the scope of the search domain.

In searching semantic multimedia it is important that the semantic space accommodates as many keywords as possible to be sure that the user’s idea is represented in that space without losing any meaning. Thus, automatic systems that extract a limited number of keywords are less appropriate. This design requirement leads us to the research area of high-dimensional spaces.

The structure of the space, i.e., the way keywords interact with each other, is defined by the distance function of that space. Distance functions are crucial in computing the semantic similarity between two multimedia documents – they define keyword independence and dependence. For example, the Euclidean distance considers keywords to be independent while graph-based metrics take keyword dependence into account. Some non-linear similarity metrics can even create semantic sub-spaces by grouping dimensions that convey the same type of information, e.g., visual concepts of information for search systems, music CD purchases for recommender systems.

In this thesis I limit the lexicon of keywords to a set of L visual and multimodal concepts that are present in images and video clips.

7.3 Keyword Vectors Computation

Data points in the keyword space correspond to a vector of keywords for a given multimedia document – the way these vectors are computed is application dependent. In some applications, keyword vectors d_W are extracted automatically from captions, Web page text, or low-level features. In this Chapter 4 and 5 we proposed a machine learning algorithm p_A that computes keyword vectors from low-level features:

$$d_W = p_A(d) = p(y | d_T, d_V) \quad (7.5)$$

The machine learning algorithm supports a large number of keywords so that the keyword space can wrap the semantic understanding that the user gives to a document. This is in line with the requirement for highly expressive descriptions of multimedia, i.e., large number of keywords.

In other type of applications, keyword vectors d_W are extracted manually from the document content by a user p_U , i.e.,

$$p_U : d \rightarrow d_W. \quad (7.6)$$

The user inspects the document to verify the presence of a concept and annotates the document with that keyword if it is present.

7.3.1 Automatic Keyword Annotations

In this section we describe how to estimate a probability function p that automatically computes the vector

$$d_W = (p(y_1 | d_f), \dots, p(y_L | d_f)), \quad (7.7)$$

of L keyword probabilities from the document's low-level features d_f . Following the approach proposed Chapter 5, each keyword w_i is represented by a naïve Bayes model. The following is a summary of Chapters 4 and 5, and is repeated here for the convenience of the reader.

Keyword Models

Keywords are modelled as text and visual data with a naïve Bayes classifier. In our approach we look at each document as a unique low-level feature vector $d_f = (f_1, \dots, f_M)$ of visual features (Section 4.1.2) and text terms (Section 4.1.3). The naïve Bayes classifier results from the direct application of Bayes law and independence assumptions between dimensions of a feature vector:

$$p(y_j | d_f) = \frac{p(y_j) \prod_{i=1}^M p(f_i | y_j)}{\sum_{i=1}^L p(y_i) p(d_f = (f_1, \dots, f_M) | y_i)} \quad (7.8)$$

Formulating naïve Bayes in the log-odds space results in

$$\log \frac{p(y_j = 1 | d)}{p(y_j = 0 | d)} = \log \frac{p(y_j = 1)}{p(y_j = 0)} + M \sum_{i=1}^M p(f_i | d) \log \frac{p(f_i | y_j = 1)}{p(f_i | y_j = 0)} \quad (7.9)$$

which casts it as a linear model that avoids decision thresholds in annotation problems.

Visual Data Processing

Three different low-level visual features are used in our implementation: marginal HSV distribution moments, a 12 dimensional colour feature that captures the histogram of 4 central moments of each colour component distribution; Gabor texture, a 16 dimensional texture feature that captures the frequency response (mean and variance) of a bank of filters at different scales and orientations; and Tamura texture, a 3 dimensional texture feature composed by measures of image coarseness, contrast and directionality. The images are tiled in 3 by 3 parts before extracting the low-level features.

Text Data Processing

Text feature spaces are high dimensional and sparse. To reduce the effect of these two characteristics, one needs to reduce the dimensionality of the feature space. We use mutual information to rank text terms according to their discriminative properties.

7.3.2 User Keyword Annotations

Professional annotations are done by experts that received some training on how to identify concepts in multimedia content, clarified all ambiguities regarding the meaning of keywords, and have no hidden intention of incorrectly annotation content. In most cases, professional annotations are obtained by a redundant voting scheme intended to remove disagreement between professional annotators. Thus, it constitutes an extra method of cleaning data annotations. Both Corel and TRECVID2005 annotations were done by experts that followed these general guidelines. Thus, we assume that professional annotations have 100% accuracy. In contrast, annotations done by real users are sometimes random, incomplete or incorrect for several reasons: the user might not be rigorous, users have different understanding of the same keyword, or it might be the result of spam annotations. In a real scenario with non-professional users one would expect to have keyword annotations with accuracies below 100%.

Following this reasoning, we use professional annotations to generate user keywords with different levels of accuracies:

- **Obtain user annotations:** generate completely accurate user keywords from the professional annotations of the collection of N multimedia documents. This corresponds to the Corel and TRECVID collections annotations.
- **Add errors to annotations:** given the professional annotations, invert a given number e of annotations which results in a classifier with an accuracy of

$$\text{accuracy} = \frac{L \cdot N - e}{L \cdot N}, \quad (7.10)$$

note that this is done to both positive and negative annotations. This step simulates different numbers of errors that users might do when annotating multimedia content.

7.3.3 Upper and Lower Bounds

Automatic annotation algorithms are not completely accurate and we do not foresee that a new algorithm will achieve a high accuracy in the near future. Thus, the user keyword annotations define the upper bound on the retrieval effectiveness that can be obtained in a search by semantic example

scenario. Correspondingly, the naïve Bayes algorithm was chosen as the automatic keyword annotation algorithm because it defines a lower bound on the retrieval effectiveness that can be obtained in a search by semantic example scenario.

7.4 Querying the Keyword Space

User queries can include keywords, multimedia examples, and arbitrary combinations of keywords and multimedia examples. The algorithm that parses the user request produces query vectors in the keyword space with the same characteristics as multimedia document vectors. For the objectives of this thesis we only need to consider single example queries. Moreover, the user request analysis algorithm must generate the query description in a fixed amount of time and with a low computational cost. This is an important feature because the system needs to answer the user request in less than one second and it should also be able to support several users simultaneously.

Thus, for each query, the system analyses the submitted example and infers a keyword vector with the automatic algorithm

$$q_W = p_A(q_f), \quad (7.11)$$

or a user provides the keywords present in the example, i.e.,

$$p_U : q \rightarrow q_W. \quad (7.12)$$

Query examples are converted into keyword vectors with the methods described in the previous section.

7.5 Keyword Vectors Dissimilarity

In this section we discuss the dissimilarity functions to compute the semantic similarity between two multimedia documents. The dissimilarity functions presented in this section assume three types of spaces: geometric, histogram-based and probabilistic spaces. Thus, all dissimilarity functions assume that either the space is linear or that keywords are independent. The computation of dissimilarity is based on functions $D(a, b)$ that are not necessarily a distance function because they might violate one of the properties of a true metric:

1. Non-negativity, $D(a, b) \geq 0$
2. Symmetry, $D(a, b) = D(b, a)$

3. Triangle inequality, $D(a, b) \leq D(a, c) + D(c, b)$
4. Identity of indiscernibles, $D(a, b) = 0$ for $a = b$

With completely accurate user keywords we isolate the dissimilarity functions from the keyword annotation process. This way we can assess how much of the semantic similarity precision is due to the keyword vector computation method and how much is due to the dissimilarity functions.

The computation of dissimilarity ranks for all documents in a database is an expensive process with linear complexity. Several methods exist to reduce this complexity, as for example sampling (Howarth and Ruger 2005b). This topic is outside the scope of this paper as we are interested in finding methods to rank documents by semantic similarity with the maximum possible precision.

7.5.1 Geometric Spaces

Geometric similarity functions operate on high-dimensional spaces and each function is implemented as a distance function under specific assumptions and/or constraints. Thus, input feature components can be any real values. However, special attention should be given to spaces with heterogeneous dimensions, e.g., metadata with discrete dimensions, that might require a specific normalization each (Gelman et al. 2003).

Minkowski Distance

The Minkowski distance between the query example q_w and a database document d_w is defined as

$$D_{\text{Minkowski}}(q_W, d_W) = L_p(q_W, d_W) = \left[\sum_{i=1}^L |q_{W,i} - d_{W,i}|^p \right]^{1/p}, \quad (7.13)$$

where the indices i concern the concept i , and p is a free parameter $p > 1$. However, Howarth and Ruger (2005a) have shown that for visual features fractional dissimilarity measures (Minkowski distance with $0.0 < p < 1.0$) offer a better performance for several types of features. In this chapter I use $p \in \{0.5, 1.0, 2.0, \infty\}$ as different distance measures. L_p is not a true metric for $p < 1$ because it violates the triangle inequality; nevertheless it can offer useful dissimilarity values. The unit spheres for $p \in \{0.5, 1.0, 2.0, \infty\}$ in the two dimensional space are illustrated in Figure 7.5.

Manhattan Distance

Manhattan distance ($p = 1.0$) corresponds to the human notion of distance between two points placed over a squared grid. The Manhattan distance is the accumulated sum of the distances

in each dimension,

$$D_{\text{Manhattan}}(q_W, d_W) = L_1(q_W, d_W) = \sum_{i=0}^L |q_{W,i} - d_{W,i}|. \quad (7.14)$$

This distance is identical to the length of shortest all paths connecting q_w and d_w along lines parallel to the coordinate system.

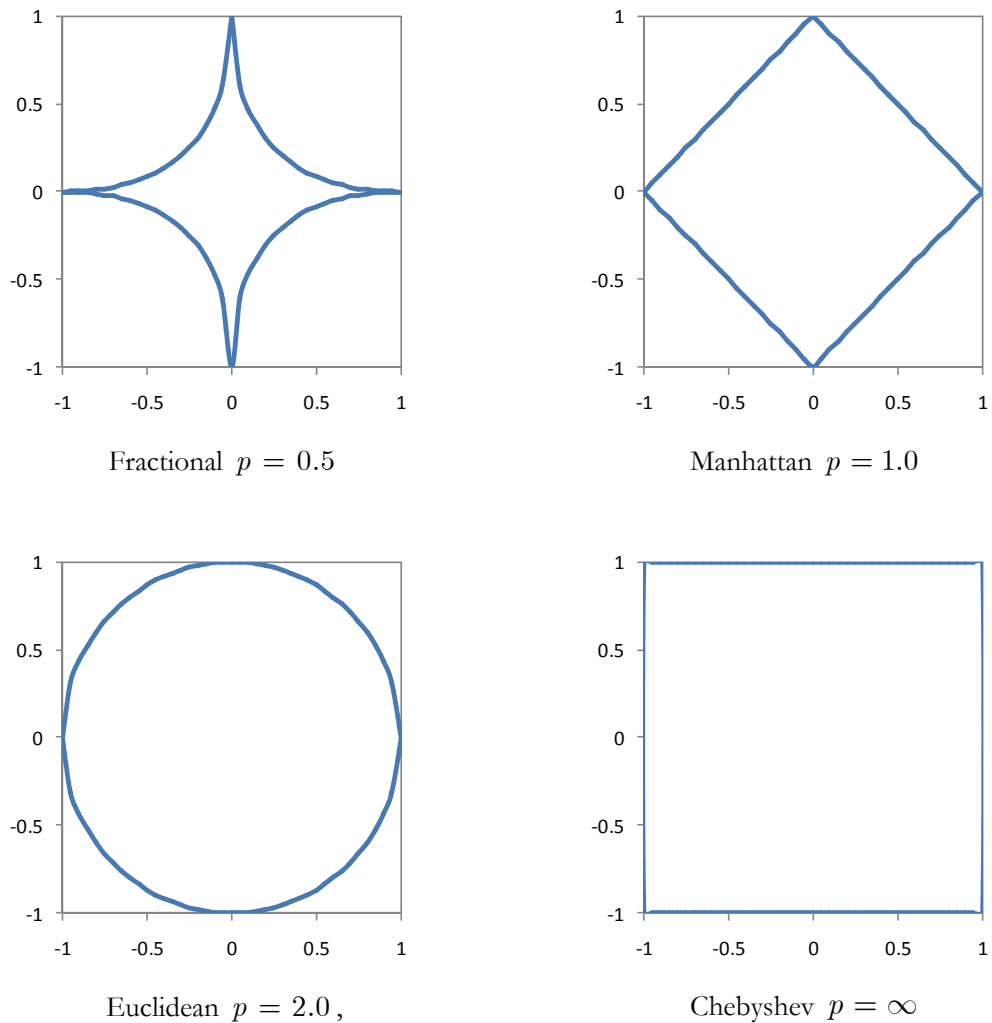


Figure 7.5. Unit spheres for standard Minkowski distances.

Euclidean Distance

Euclidean distance (Minkowski distance with $p = 2.0$) corresponds to the human notion of distance between two points in a real coordinate space, expressed as

$$D_{\text{Euclidean}}(q_W, d_W) = L_2(q_W, d_W) = \sqrt{\sum_{i=0}^L (q_{W,i} - d_{W,i})^2}. \quad (7.15)$$

Chebyshev

The Chebyshev distance (Minkowski distance with $p = \infty$) measures the maximum of the distance in each dimension. It is expressed as

$$D_{\text{Chebyshev}}(q_W, d_W) = L_\infty(q_W, d_W) = \max_{0 \leq i \leq L} |q_{W,i} - d_{W,i}|. \quad (7.16)$$

Cosine Distance

Since we work in high-dimensional spaces, in geometric terms one can define the independence between two vectors as the angle between them. This gives an indication as to whether two vectors point to a similar direction or not. This is the well known cosine similarity which becomes a dissimilarity by taking the difference to 1:

$$D_{\text{Cosine}}(q_W, d_W) = \cos(q_W \angle d_W) = 1 - \frac{q_W \cdot d_W}{\|q_W\| \cdot \|d_W\|} \quad (7.17)$$

Geometric correlation is one of the several possible ways to measure the independence of two variables. Also, the cosine distance is a special case of Pearson correlation Coefficient when data are normalized with mean zero.

7.5.2 Histograms

Histograms are computed by discretizing feature spaces into bins, meaning the proportion of cases in which this bin is occupied. Histograms are widely applied in colour spaces where each bin corresponds to a given segment of the colour space measuring the proportion of pixels that fall into that segment. In our scenario we consider one concept to be equivalent to one bin of the histogram.

Canberra Distance

The Canberra distance is the sum over the difference in each bin normalized by the sum of the corresponding bin sizes:

$$D_{\text{Canberra}}(q_W, d_W) = \sum_{i=1}^L \frac{|q_{W,i} - d_{W,i}|}{|q_{W,i}| + |d_{W,i}|}. \quad (7.18)$$

This distance has been used before with relative success in low-level-feature based image retrieval (Kokare, Chatterji and Biswas 2003).

Histogram Intersection

Histogram intersection is a measure that was applied in the early 1990s (Swain and Ballard 1991) as a method to index images by colour. This distance measures what two histograms have in common by computing their intersection. The distance is normalized with the size of the smaller histogram:

$$D_{\text{HistInt}}(q_W, d_W) = \frac{\sum_{i=1}^L \min(q_{W,i}, d_{W,i})}{\min(|q_W|, |d_W|)} \quad (7.19)$$

This measure is equivalent to L_1 distance.

7.5.3 Probabilistic Spaces

In this section I describe statistics based measures of similarity: divergences between two probability density distributions, and the likelihood that two samples of a given population came from the same probability density distribution.

Kullback-Leibler Divergence

In statistics and information theory the Kullback-Leibler (KL) divergence is a measure of the difference of two probability distributions. It is the distance between a “true” distribution (the query vector) to a “target” distribution (the document vector). The KL divergence is defined as

$$D_{\text{KL}}(q_W \parallel d_W) = \sum_{i=1}^L p(q_{W,i}) \log \frac{p(q_{W,i})}{p(d_{W,i})} \quad (7.20)$$

In information theory it can be interpreted as the expected extra message length needed by using a code based on the candidate distribution (the document vector) compared to using a code based on the true distribution (the query vector). Note that the KL divergence is not a true metric as it is not symmetric.

Jensen-Shannon Divergence

The Jensen-Shannon (JS) divergence is the symmetrised variant of the KL divergence and provides a true metric to compare two probability distributions:

$$D_{\text{JS}}(q_W, d_W) = \frac{1}{2} D_{\text{KL}} \left(q_W \parallel \frac{1}{2}(q_W + d_W) \right) + \frac{1}{2} D_{\text{KL}} \left(d_W \parallel \frac{1}{2}(q_W + d_W) \right), \quad (7.21)$$

An interesting characteristic of the JS divergence is that one can assign different weights to each

distribution (Lin 1991). This makes it particularly useful for decision problems where weights could be the prior probabilities.

Pearson Correlation Coefficient

The Pearson correlation coefficient between two random variables is a measure of the strength of their independence. It measures the degree of linear relationship between two random variables. Positive values of correlation indicate a linear relationship, while negative values correspond to a negative linear relationship between the variables. A correlation of 0 corresponds to the case where the variables are independent. The Pearson correlation coefficient expressed as

$$D_{\text{Corr}}(q_W, d_W) = \frac{\text{cov}(q_W, d_W)}{\sigma_{q_W} \cdot \sigma_{d_W}} = \frac{\sum_{i=1}^L (q_{W,i} - \overline{q_W})(d_{W,i} - \overline{d_W})}{\sqrt{\left(\sum_{i=1}^L (q_{W,i} - \overline{q_W})^2\right) \cdot \left(\sum_{i=1}^L (d_{W,i} - \overline{d_W})^2\right)}}, \quad (7.22)$$

is equivalent to the cosine distance when both variables are normalized to mean zero.

7.6 Evaluation

I have focused experiments on similarity ranking of a single semantic-multimedia example (search by semantic example) in this chapter. The goal is to study the characteristics of the semantic space and how it behaves with different parameters.

7.6.1 Collections

I carried out experiments on similarity ranking of semantic multimedia using an image collection and a multimodal collection. Both collections were split into training and test set, and they have two levels of annotations: one used to build the models of keywords that correspond to the lexicon of keywords of the keyword space; and a second level of categories that correspond to a particular category. More details regarding these collections are provided in Chapter 2.

	Training Examples	Test Examples	Keywords	Categories
Corel Images	4,500	500	260	50
TRECVID	23,709	12,054	39	8

Table 7.1. Summary of collections used on the experiments.

Corel Images

The collection is split into a training set of 4,500 images and a test set of 500 images. Each

image is annotated with one to five keywords from a vocabulary of 371 keywords. Only keywords with at least one image both in the test and training set were used, which reduces the size of the vocabulary to 260 keywords. The collection is already organized into 50 image categories, such as *rural France*, *Galapagos wildlife* and *nesting birds*, as illustrated in Figure 7.24.

TRECVID

To test semantic similarity on video data we used the TRECVID data: since only the training set is completely labelled, we randomly split the English training videos into 23,709 training documents and 12,054 test documents. Key-frame keywords have two origins: the standard vocabulary of 39 keywords provided by NIST, plus the large-scale LS-COMM ontology of 400 keywords provided by Naphade et al. (2006). We trained the keyword models on the 39 keywords to form the keyword space and used 8 categories as relevance judgments (ground truth) for evaluation (*landscape, weapons, politics, vehicle, group, daytime outdoor, dancing* and *urban park*). The 8 categories were selected from the LS-COMM ontology as non overlapping keywords with the other 39 keywords and had an enough number of examples. Note that because TRECVID categories are not annotated at the level of groups of documents we expect to have a lower accuracy in TRECVID when compared to Corel that have meaningful categories.

7.6.2 Experiments Design

Before proceeding to the keyword space dissimilarity evaluation experiments we first learned the naïve Bayes keyword models on the training set of each collection. Dissimilarity evaluation is done on the collections test set and with the corresponding keyword models. Each individual test example is used as a query example to rank the remaining test examples by semantic-similarity. Formally, the followed methodology is described next:

1. Learn the naïve-Bayes model for each keyword on the training set of each collection (260 models for Corel and 39 for TRECVID). Note that we do not reuse the training set as the search database in contrast to (Rasiwasia, Moreno and Vasconcelos 2007).
2. Submit a test document as a query example to rank the remaining test examples by semantic similarity.
3. Compute keyword annotations for both documents and query with the different algorithms:
 - a. Automatic keywords with the naïve-Bayes algorithm (260 keywords for Corel and 39 for TRECVID).
 - b. User keywords with varying accuracy.

4. Rank documents by their semantic similarity to the query example according to a given dissimilarity function:
 - a. Minkowski (0.5), Manhattan, Euclidean, Chebychev, Cosine, Canberra, Kullback-Leibler and Jensen-Shannon.
5. The category of the query example is used as relevance judgment to evaluate the rank of documents.
6. Repeat steps 2 to 5 for all test examples.

The above methodology is repeated for each dissimilarity function, dataset, and keyword vector computation algorithm. This way we isolate the variables of the problem that we are interested in studying: semantic-similarity functions, influence of user annotations accuracy, influence of the number of keywords.

Average precision, mean precision at 20, and interpolated precision-recall curves are the measures used for comparing the different ranks. Mean average precision (MAP) and mean precision at 20 (MP@20) are the measures to evaluate the different systems. Conceptually, average precision is the area under the precision recall curve. Average precision as a performance measure has the advantage that it gives a greater weight to results retrieved early. Mean precision at 20 is an important measure of the usability of the system as many users only look at the top results, see (iProspect April 2006).

7.6.3 Results and Discussion

Automatic Keywords

These results are obtained with the output of the naïve Bayes classifier and for the keyword space with the maximum number of keywords. Thus, it evaluates the dissimilarity functions in a fully automated scenario.

The MAP obtained with Canberra and Cosine was consistently better than the others as we can see from Figure 7.6 for the Corel collection and from Figure 7.10 for the TRECVID collection. The same behaviour could be observed with the MP@20, Figure 7.7 and Figure 7.11 for the Corel and TRECVID collections respectively. As expected the difference between the same family of metrics was not great: Minkowski based metrics are all similar (apart from Chebyshev, $p = \infty$); probabilistic based metrics (KL and JS divergence) are all in the same range. However, note that in all situations retrieval performance is always well above the random rank, the random MAP is 0.034 for the Corel collection and 0.029 for the TRECVID collection.

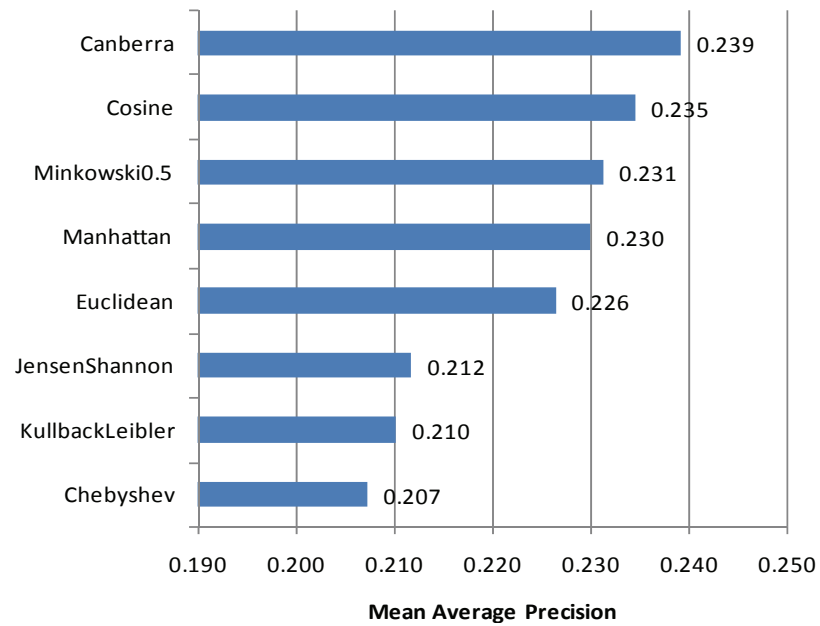


Figure 7.6. MAP of the different dissimilarity functions (Corel Images).

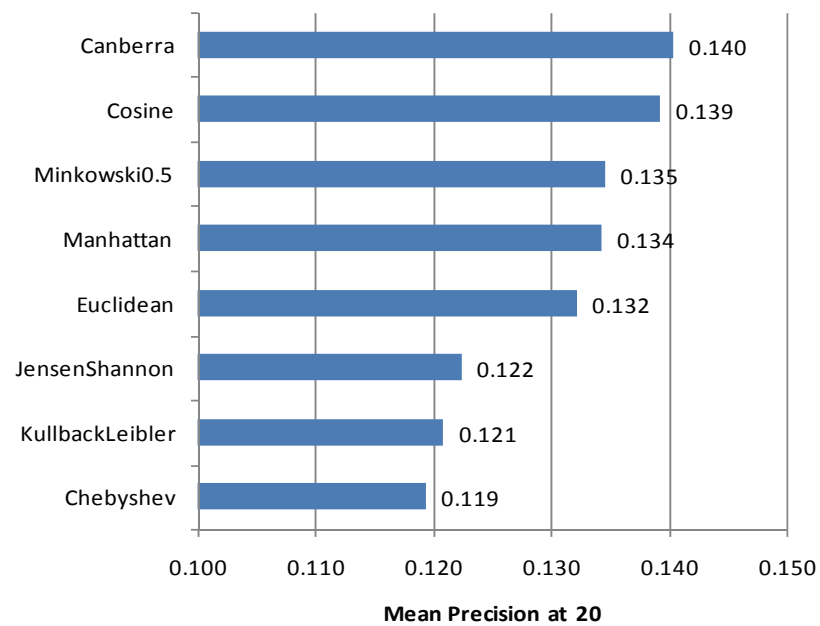


Figure 7.7. MP@20 of the different dissimilarity functions (Corel Images).

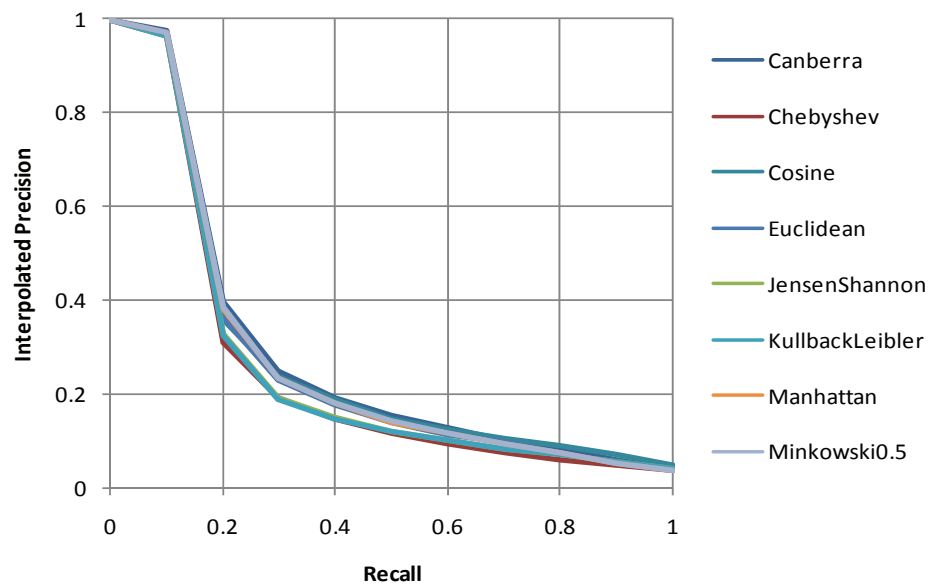


Figure 7.8. Interpolated precision-recall curves of the different dissimilarity functions (Corel).

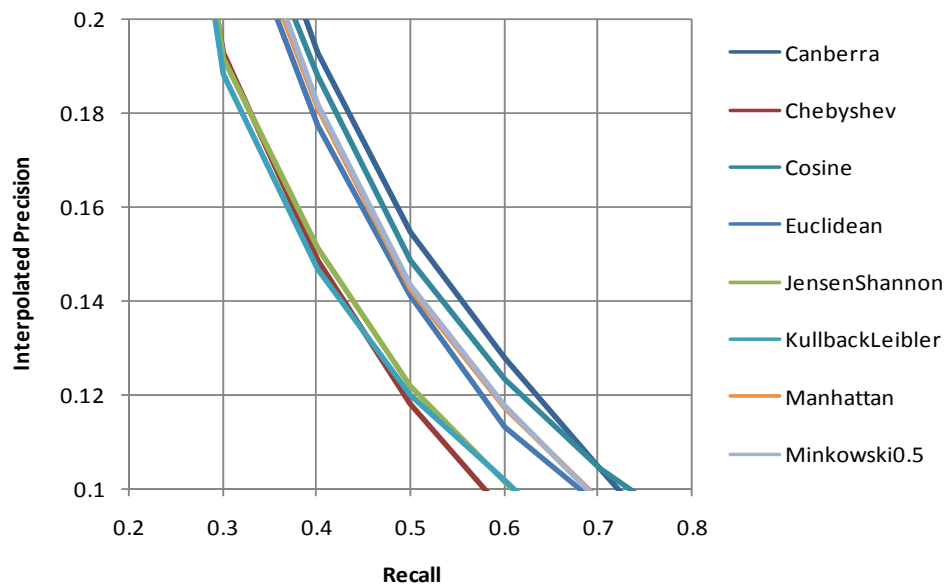


Figure 7.9. Interpolated precision-recall curves of the different dissimilarity functions (Corel).

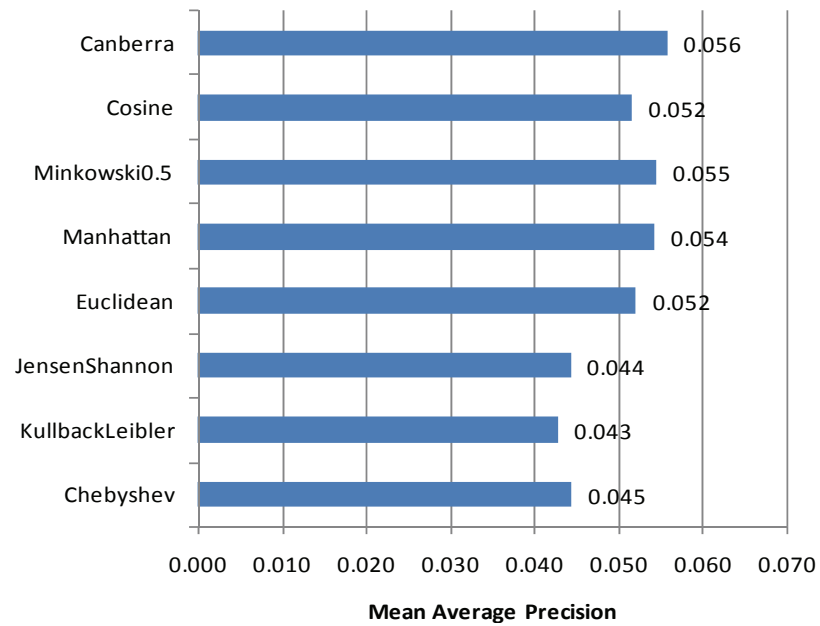


Figure 7.10. MAP of the different dissimilarity functions (TRECVID).

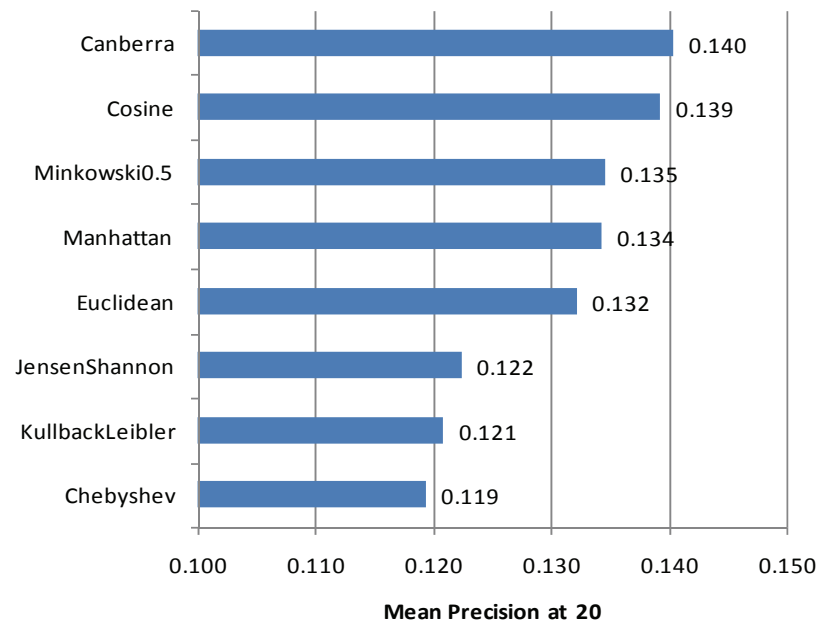


Figure 7.11. MP@20 of the different dissimilarity functions (TRECVID).

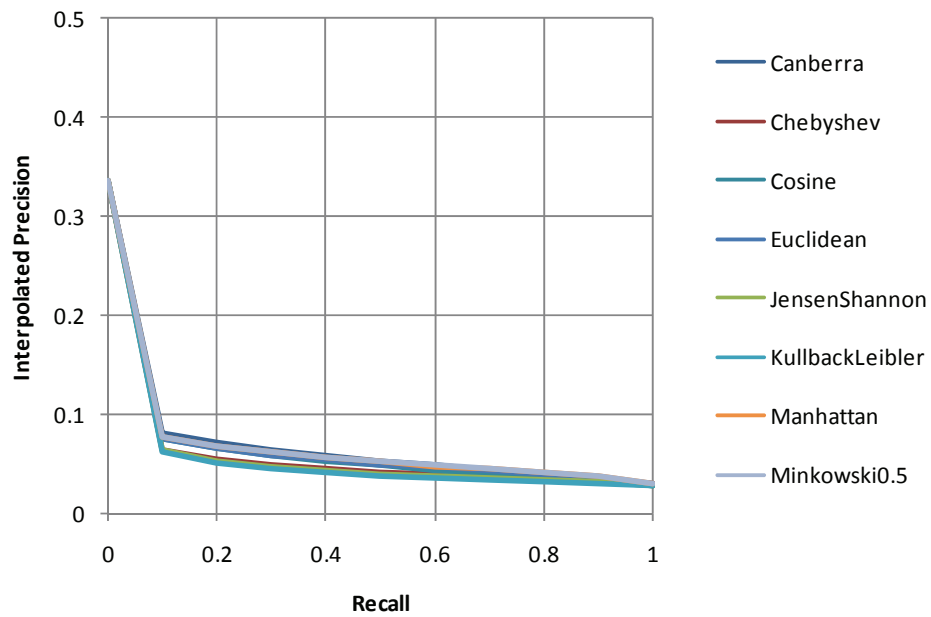


Figure 7.12. Interpolated precision-recall curves of the different dissimilarity functions (TRECVID).

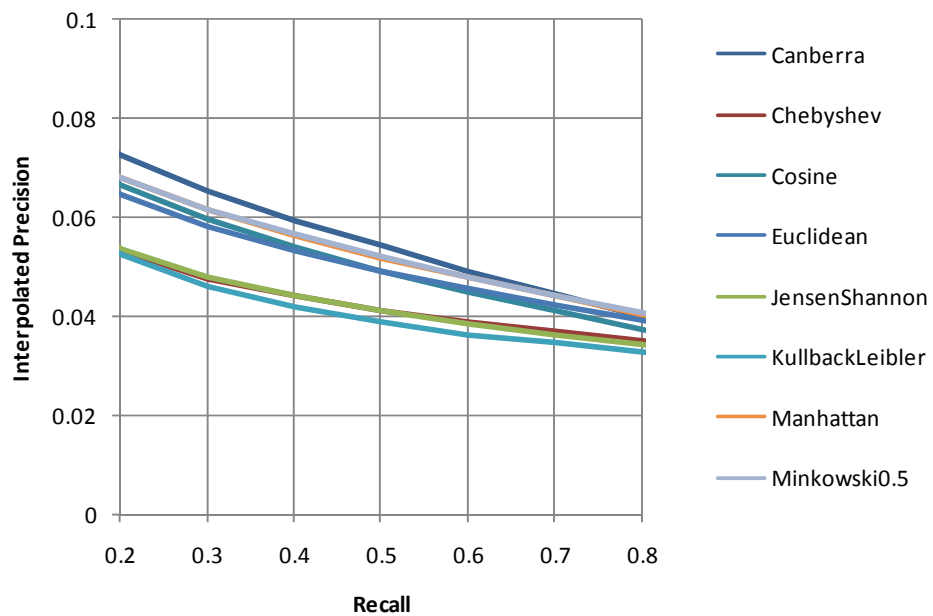


Figure 7.13. Interpolated precision-recall curves of the different dissimilarity functions (TRECVID).

A more careful analysis of the different dissimilarity functions with automatic keywords is provided by the precision-recall curves in Figure 7.8 for the Corel collection and in Figure 7.12 for the TRECVID. These curves confirm that dissimilarity functions show a common pattern and they are grouped according to their family. In both collections, curves are more different from each other in the recall interval between 0.20 and 0.80 (Figure 7.9 and Figure 7.13 provide zoom over this range of recall). In these graphs the grouping of the functions according to their type is more noticeable. Note that Canberra and Cosine dissimilarity functions are very similar along the entire curve.

User Keywords

The evaluation presented in this section creates a keyword space with the output of professional level user keywords according to the procedure described in section 7.3.2. Thus, it allows assessing how different dissimilarity functions behave in the presence of user generated annotations.

The user keyword results provide us with the upper bound of the retrieval effectiveness. The upper bound is obviously dependent on the similarity function. Figure 7.14 illustrates the precision-recall curves of the different dissimilarity functions on the Corel collection. It is visible the improvement over automatic keywords for all dissimilarity functions. Figure 7.15 illustrates the same experiment on the TRECVID collection.

Dissimilarity	Corel Images	TRECVID
Minkowski (0.5)	0.387	0.092
Manhattan	0.435	0.096
Euclidean	0.435	0.097
Chebyshev	0.240	0.073
Cosine	0.464	0.131
Canberra	0.174	0.055
Kullback-Leibler	0.415	0.092
Jensen-Shannon	0.460	0.123

Table 7.2. MAP for user keywords.

Table 7.2 summarizes the MAP values for both collections. The most noticeable fact is that even with completely accurate annotations we cannot pass a value of 50% MAP. These results allow us to draw many speculations and are a good source of many new research questions. There is an obvious gap between the annotated keywords and the unknown categories. Note that this is different from the notion of semantic gap between low-level features and concepts. It is actually a gap among concepts, in this case between the annotated concepts and the user information need.

This point to two possible solutions: increase the number of concepts or investigate semantic spaces to represent multimedia information and possible similarity metrics. The first solution is the simple application of brute force, hoping to have comprehensive annotations with better high-level concept extractors. The second solution suggests investigating similarity functions that incorporate concept interdependencies and are robust to noisy document descriptions.

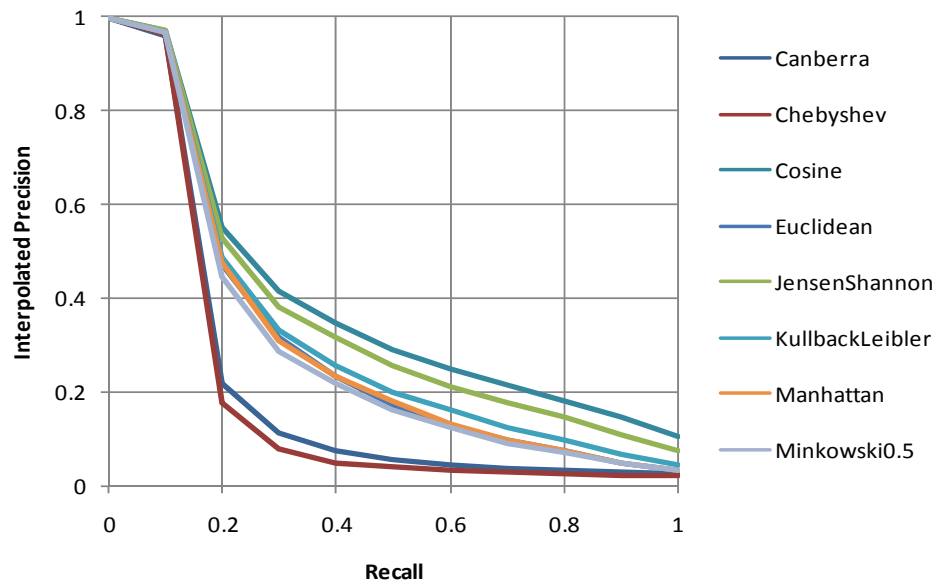


Figure 7.14. Interpolated precision-recall curves of the different dissimilarity functions (Corel).

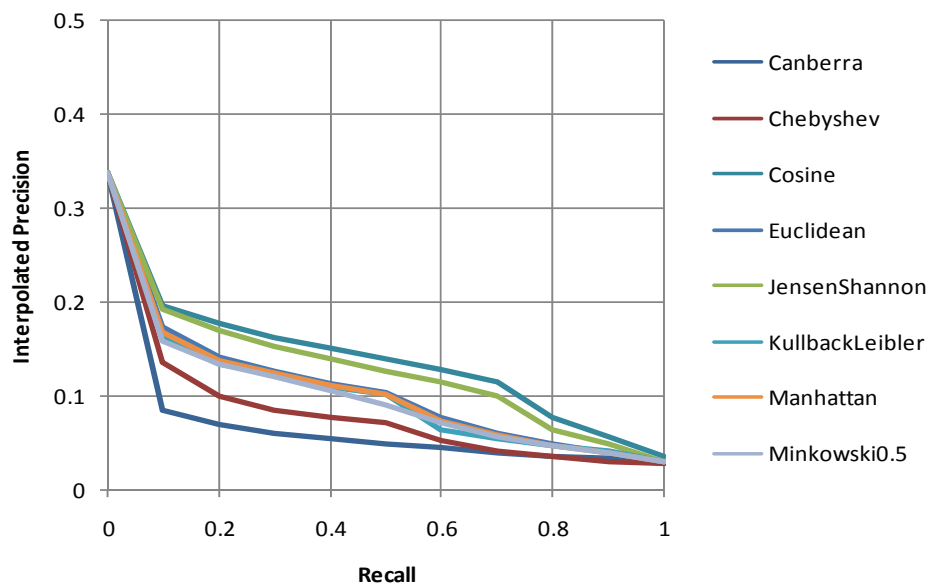


Figure 7.15. Interpolated precision-recall curves of the different dissimilarity functions (TRECVID).

User Keywords Accuracy

In this experiment we study the influence of the accuracy of annotations done by users on the retrieval MAP. The procedure described in section 7.3.2 was followed to evaluate the retrieval effectiveness with varying levels of annotation accuracy. This procedure tries to simulate the effect of incorrect user annotations that might occur for different reasons, e.g., interpretation of keyword, spam, or incomplete annotations.

In the Corel collection, both MAP and MP@20 are extremely sensitive to small changes in highly accurate user keywords as can be seen in Figure 7.16 and Figure 7.17 respectively. There is a major change in retrieval precision when classifiers accuracies go from 90% to 100% and it is relatively stable for accuracies under 90%. TRECVID exhibits the same MAP and MP@20 trend in Figure 7.18 to Figure 7.19 respectively: both MAP and MP@20 suffers an exponential growth with the increase of user keywords accuracy. Note that retrieval effectiveness in Corel is more sensitive to the classifiers accuracy than with TRECVID. This is a consequence of the higher correlation between the Corel keywords and categories than in TRECVID.

A more detailed analysis of the curves shows that for the same level of retrieval effectiveness, MP@20 needs more accurate classifiers than the MAP. This is not a surprise as MP@20 measures the specialization of a retrieval system, thus, more accurate classifiers allow the system to be better at retrieving particular documents at the top of the rank.

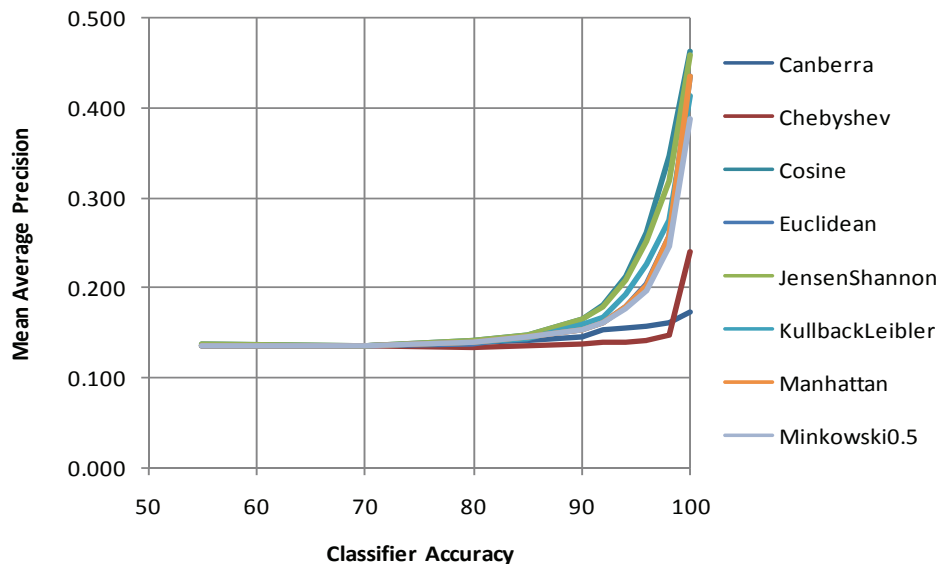


Figure 7.16. Effect of user keywords accuracy on the MAP (Corel).

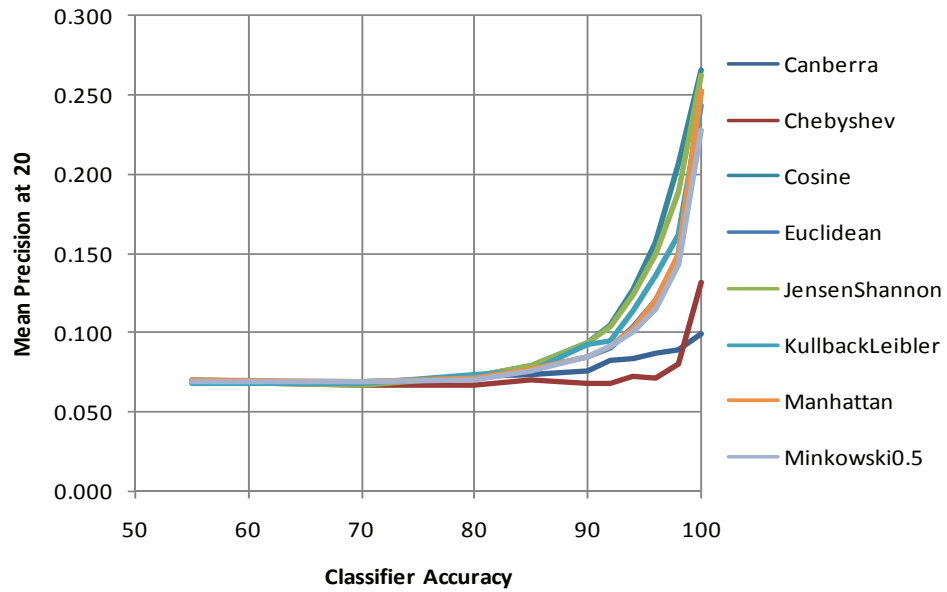


Figure 7.17. Effect user keywords accuracy on the MP@20 (Corel).

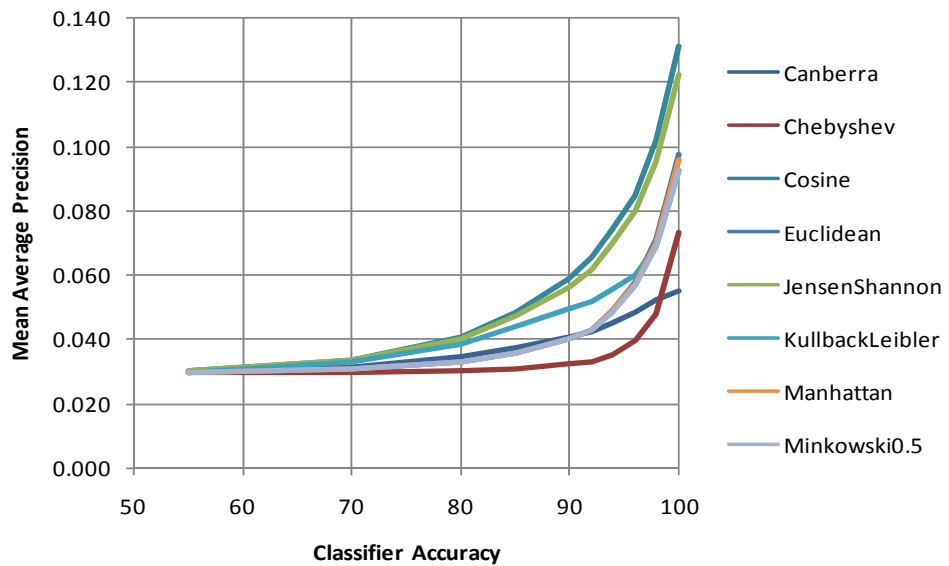


Figure 7.18. Effect of user keywords accuracy on the MAP (TRECVID).

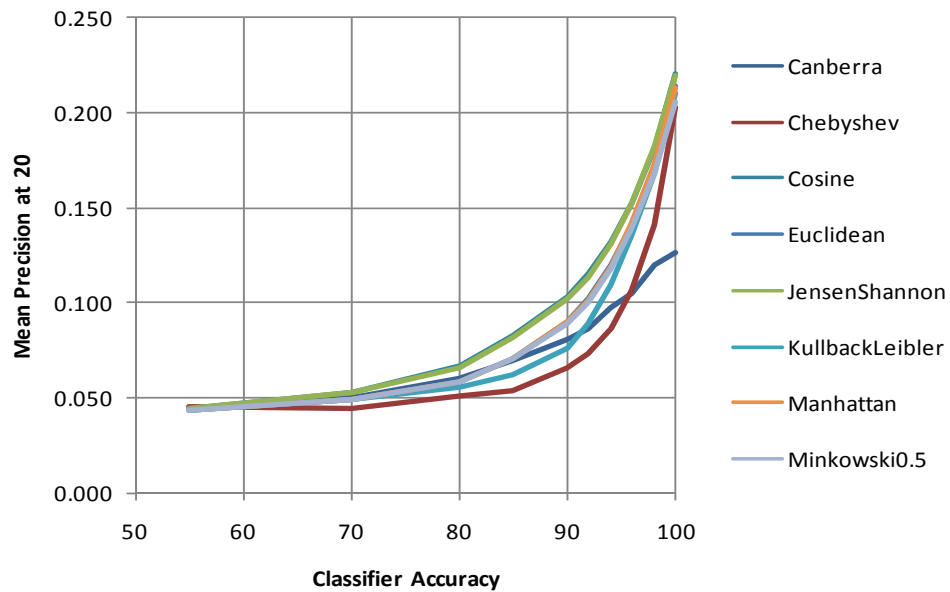


Figure 7.19. Effect of user keywords accuracy on the MP@20 (TRECVID).

User Keywords versus Automatic Keywords

The MAP upper bound of retrieval by semantic similarity is computed with completely accurate user keywords. This bound is obtained with professional user keyword annotations. In the Corel collection the upper bound is 0.464; in the video-clip collection the upper bound is 0.131. Table 7.3 provides a summary that allows comparing automatic keywords to user keywords. The summary highlights the retrieval MAP for 100% and 95% accurate user keywords (these values are taken from curves on Figure 7.16 and Figure 7.18).

In the Corel collection, a retrieval scenario with automatic keywords and the Cosine dissimilarity results in a MAP of 0.235, which is on par with or even slightly better than the corresponding one of the 95% correct user keyword (0.226). The same holds for other similarity functions. Thus, automatic keywords have roughly the same performance as 95% correct user keywords. Experiments on the TRECVID collection displays an equivalent trend as can be inferred from Table 7.3: the MAP of the automatic keywords 0.054 is on par with the corresponding MAP of the 95% correct user keywords (0.051).

The encouraging news here is that we are comparing a simple automatic annotation algorithm to professional level annotations, and one would expect there to be scope for improvement.

Dissimilarity	Corel Images			TRECVID		
	Automatic keywords	User keywords (100%)	User keywords (95%)	Automatic keywords	User keywords (100%)	User keywords (95%)
Minkowski (0.5)	0.231	0.387	0.197	0.056	0.092	0.056
Manhattan	0.230	0.435	0.204	0.054	0.096	0.057
Euclidean	0.226	0.435	0.205	0.052	0.097	0.058
Chebyshev	0.207	0.240	0.141	0.045	0.073	0.039
Cosine	0.235	0.464	0.226	0.052	0.131	0.084
Canberra	0.239	0.174	0.157	0.056	0.055	0.048
Kullback-Leibler	0.210	0.415	0.224	0.043	0.092	0.060
Jensen-Shannon	0.212	0.460	0.230	0.044	0.123	0.080
Random		0.034			0.029	

Table 7.3. Comparison between automatic keywords and user keywords.

Keyword Space Dimensionality

In the previous evaluations the keyword space included the full range of concepts, independently of their value to the ranking process. This affects accuracy as some of the concepts are either noise or are irrelevant to most searches. In this section we study the effect of removing noisy keywords from the keyword space in the ranking process with automatic keywords.

The keyword space is built by progressively adding keywords according to their precision. Thus, the keywords with higher average precision are added first. This is similar to unsupervised feature selection that is exclusively based on the accuracy of the keywords. Thus, I do not use the category to select the keywords (e.g., use the final objective to select dimensions like in normal feature selection) because in this experiment one should not know the category beforehand.

In the Corel collection we can observe that the first concepts carry more information value. As lower precision keywords are added to the semantic space the MAP (Figure 7.20) and the MP@20 (Figure 7.21) also increase. It is important to note the robustness to noise that this experiment illustrates: Canberra, Correlation and Cosine continue to show a good robustness to noise.

The same general conclusions can be drawn from the MAP (Figure 7.22) and MP@20 (Figure 7.23) curves on the TRECVID collection. However, in this dataset we see that the relation between the keywords and the categories is not as clear as in the Corel collection. This is probably due to the fact that Corel keywords/query categories were done by trained professionals with the explicit intention of creating meaningful hierarchy (keywords/ categories) and TRECVID keywords/ categories were done with the sole purpose of annotating content. Also, note that the MP@20 is more sensitive than MAP to the initial keywords with higher precision. The reason for this is that MP@20 looks at a limited set of results, thus making the first accurate keywords more valuable.

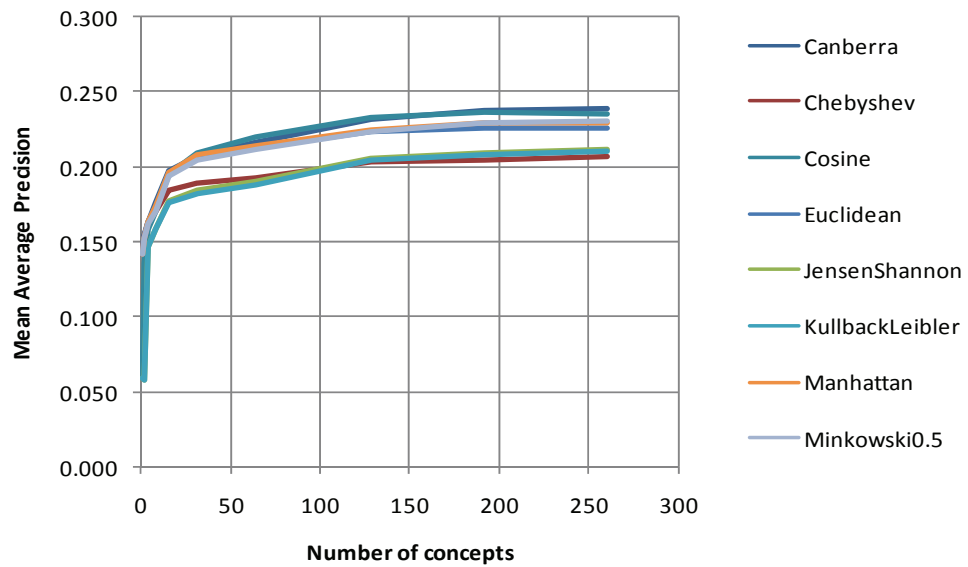


Figure 7.20. Effect of the number of concepts on the MAP (Corel).

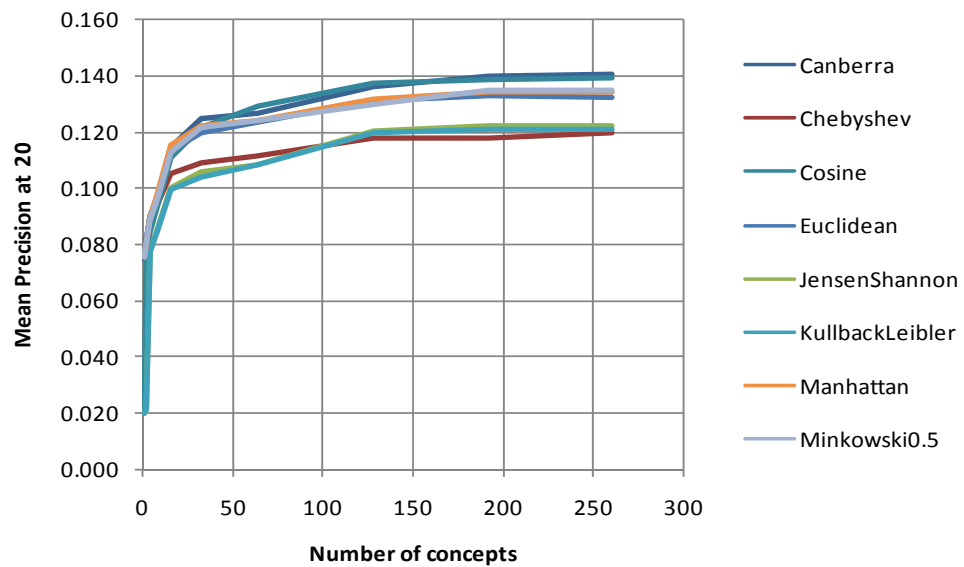


Figure 7.21. Effect of the number of concepts on the MP@ 20 (Corel).

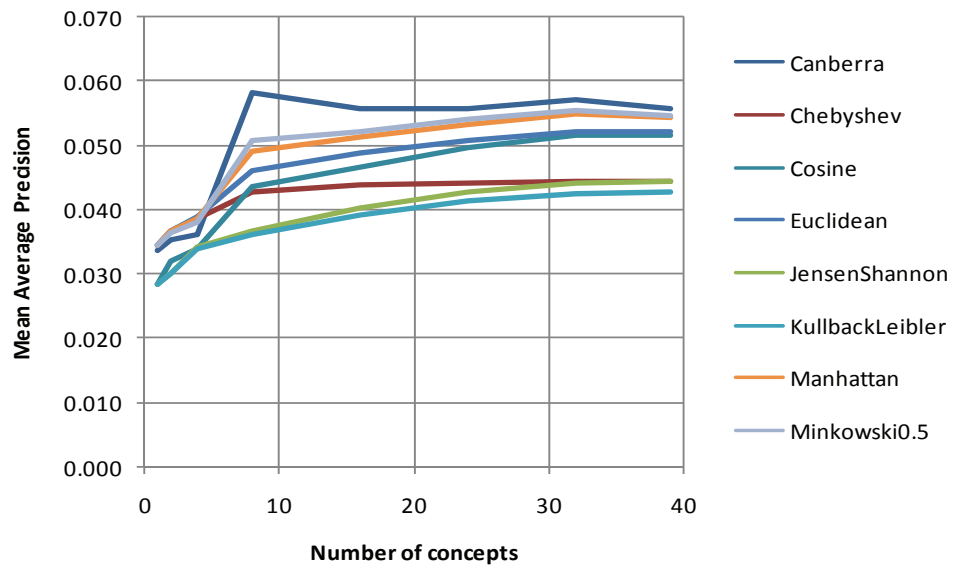


Figure 7.22. Effect of the number of concepts on the MAP (TRECVID)

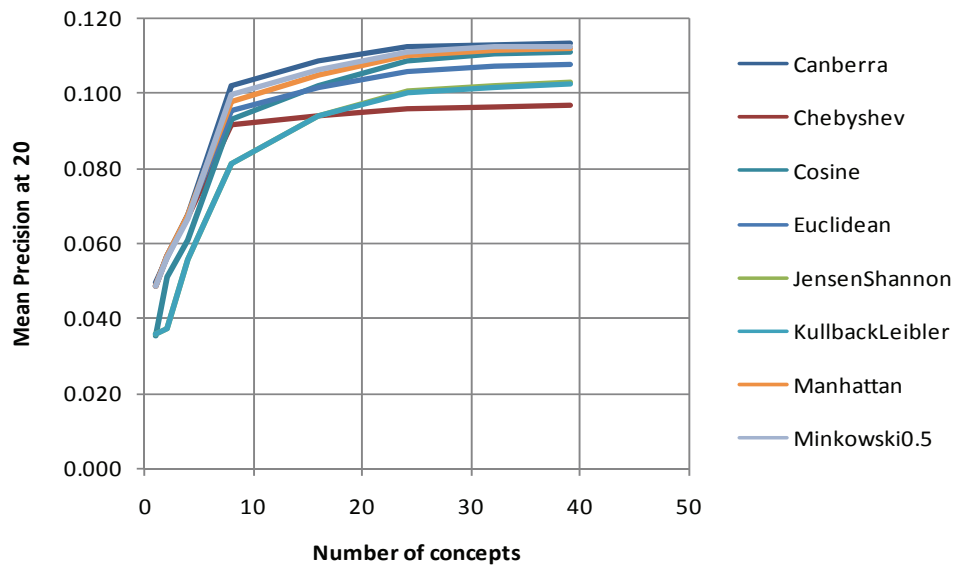


Figure 7.23. Effect of the number of concepts on the MP@20 (TRECVID)

Uncontrolled Vocabularies

Non-professional users annotate content with every keyword that they wish. This generates uncontrolled vocabularies nowadays called *folksonomies*. Their advantages are obvious from the multitude of social-media Web applications that apply it successfully.

However, the uncontrolled nature of folksonomies causes many problems in the computation of semantic dissimilarities between two multimedia documents. First, it is never possible to know the correct meaning that a user gives to a keyword, e.g., the keyword football means different sports for different cultures. Second, the user might dishonestly annotate a document with a popular keyword to attract other users. Third, users might have different criteria to annotate documents, e.g., some users might rigorously annotate all keywords while others might skip the obvious ones. Thus, uncontrolled vocabularies offer an good solution to the problem of multimedia annotation but is not a solution that delivers 100% accuracy.

With automatic methods these problems do not exist: algorithms' errors are always consistent for the same type of content, e.g., similar content suffer the same type of annotations noise. Thus, we believe that the results of the proposed framework show that automatic methods have an important role in the semantic exploration of multimedia content.

Query Processing Scalability

The time complexity of the semantic query analysis is a crucial characteristic that I consider to have the same importance as precision. For this reason I carefully chose algorithms that can handle multimedia semantics with little computational complexity. Table 7.4 illustrates the times required to extract the visual-features and to run the semantic-multimedia analysis algorithm. Measures were done on an AMD Athlon 64 running at 3.7GHz. Note that these values are for the inference phase and not for the learning phase.

Task	Time (ms)
margHSV 3x3 feature	30
Tamura 3x3 feature	54
Gabor 3x3 feature	378
Semantic annotation (260 concepts)	9

Table 7.4. Semantic analysis performance per image.

These times can be further improved because all intermediate steps are written onto disk which takes much more time than the algorithm itself. On a production system, the data generated from analysing a query example would not have to be written to disk thereby improving the computational performance.

Semantic Relevance

Assessing the user information needs from an example is always a difficult task. In this chapter I assumed that the information need can be represented by a set of keywords extracted from the example and evaluated with categories. The commonly used measures of precision and recall use a binary relevance model to identify relevant and non relevant documents. However, in the current scenario the relevance of a document is difficult to measure because semantic relevance is gradual and contextual. The problem is even more complex for several reasons, e.g., for a particular query an image with one matching keyword might be more meaningful than an image with two matching keywords; an image might belong to different categories but only one category is the required one. Figure 7.24 illustrate some of these situations: the images on the *rural-France* category can illustrate other unknown categories, e.g., *old-buildings*; images on the categories *nesting birds* and *Galapagos wildlife* overlap semantically in many aspects.










Image category: rural France	Image category: Galapagos wildlife	Image category: nesting birds
 <p data-bbox="405 1184 571 1209">barn;buildings;field;</p>	 <p data-bbox="826 1184 925 1209">crab;rocks;</p>	 <p data-bbox="1203 1184 1331 1209">birds;nest;tree;</p>
 <p data-bbox="405 1453 571 1478">buildings;field;tree</p>	 <p data-bbox="794 1453 960 1478">giant;rocks;tortoise</p>	 <p data-bbox="1171 1453 1353 1478">barn;birds;nest;wood;</p>
 <p data-bbox="405 1722 571 1747">castle;hills;sky;stone</p>	 <p data-bbox="831 1722 917 1747">birds;nest</p>	 <p data-bbox="1193 1722 1331 1747">birds;nest;water;</p>

Figure 7.24. Example of image keyword-categories relationships.

This is a consequence of the two problems of semantic relevance judgments: incompleteness and type of relevance judgment. Incompleteness of relevance judgments derives from the fact that not all labels present in a document are marked as present. The second problem concerns the type

of relevance judgments (keywords and categories) used in these experiments. Thus, ranked relevance where documents are ordered by similarity is more adequate to investigate functions for semantic similarity. Note that, although binary relevance judgments are an approximation to this ideal situation, they still provide a good research setup.

7.7 Conclusions and Future Work

This chapter addressed the problem of exploring multimedia by semantic similarity in a keyword space. Managing multimedia by their keywords and categories is a complex task involving a long chain of information processing algorithms. We presented experiments to analyze three aspects of the process: (1) the influence of the accuracy of user keyword annotations versus automatic keyword annotation algorithms and (2) functions to compute semantic similarity; and (3) the dimensionality of the keyword space.

Our evaluation allows us to draw the following conclusions regarding multimedia semantic similarity:

- A keyword space defines a new feature space that needs to be further investigated
- Automatic keyword annotations perform better than 95% accurate user keywords but is still below completely accurate user keywords
- User keywords show that similarity is highly sensitive to extremely accurate keyword annotations (the difference between 95% and 100% correct keywords)
- User keywords show that similarity is robust to errors for averagely accurate keyword annotations (range between 80% and 95%)
- All considered dissimilarity functions perform similarly
- The increase of the semantic space dimensionality, results in a corresponding increase in retrieval effectiveness

We believe that the results of the proposed framework show that automatic methods have an important role in the semantic exploration of multimedia content. These conclusions together with the experiments results shed some light on the problem of semantically comparing two multimedia documents.

7.7.1 Future Work

The presented framework is highly flexible and raises many research questions and hypothesis – experiments on this chapter only researched a small part of the problem. Thus, the research questions that I consider more relevant as future work are:

- **Graph-based semantic similarity:** There is a gap between the high-level concepts and the high-level search categories. Not all categories can be represented by those concepts. This point to the hypothesis that search categories are not arranged in a sphere type pattern but with some other pattern, e.g., a disjoint mixture of clusters. Graph-based functions should be able to embedded concept dependencies in the computation of semantic dissimilarity.
- **Human ranked relevance judgments of semantic similarity** would provide a better source to investigate other measures of semantic similarity.
- **Evaluation of queries with arbitrary combinations of keyword and semantic examples:** The presented experiments were all unsupervised retrieval, however a human user can provide more information, e.g., keywords, to guide the retrieval model towards the correct search topic.
- **Extension of the search by semantic-example paradigm to other types of data:** The collections in this evaluation cover images and videos. Other types of information such as Web or cross-media documents could be included.

8.1 Achievements

The aim of this thesis is to research statistical models of semantic-multimedia information with a view to enhance multimedia retrieval applications. In the first part I proposed an approach that creates a layer linking low-level multimedia data to semantic information in the form of keywords. These links were established by the keyword modelling framework proposed in Chapters 4 and 5:

- The first step of the framework builds a multi-modal feature space that is selected by the MDL principle. The optimal feature space is selected from a group of candidate models computed with a hierarchical EM algorithm for visual data and the mutual information criterion for text data (Chapter 4).
- Keywords are then modelled with an algorithm from the family of linear models: Rocchio classifier, naïve Bayes, and logistic regression with L_2 regularization (Chapter 5).

This framework was thoroughly evaluated with different collections in search-by-keyword and search-by-semantic-example scenarios that illustrated its excellent trade-off between flexibility, scalability and precision:

- **Flexibility:** the framework supports both single-media and multi-modal information (Chapter 4), and a large number of keywords (Chapter 5).
- **Scalability:** the framework can easily scale in terms of computational complexity and number of keywords (Chapter 7).

- **Effectiveness:** retrieval results of keyword models are slightly below but in the same range of the best algorithms (Chapter 5).

In the second part of this thesis we addressed the problem of searching semantic-multimedia. The most relevant contribution of this second part is the proposal and the thorough characterization of a keyword space to represent semantic-multimedia. From these experiments the most outstanding findings are:

- Search by semantic-example offer excellent results, i.e., it was a surprise to find that multimedia semantic similarity with automatic keywords performs as good as or better than 95% accurate user keywords (Chapter 7).
- All considered dissimilarity functions perform similarly and the increase in the keyword space dimensionality results in an improvement of the retrieval effectiveness (Chapter 7).

This research gave us new ideas to improve the quantitative results of the algorithms we evaluated. In Section 8.2 we will discuss the most relevant ones. Last of all, my quantitative results constitute a stable base to carry out a qualitative inspection and analysis.

On the qualitative side we were able to gather a critical view of (a) the research done throughout the course of this thesis and (b) the research done by the multimedia information retrieval community (Chapter 3 and 6). The most relevant qualitative achievement is a better understanding of how to exploit semantics in multimedia retrieval systems by processing it at the two extremes of the information chain: at the content side and at the user side. This research direction turned out to be very relevant as most research in the past was dedicated to the development of multimedia analysis exclusively on the content side (Forsyth 2001). Taking this to the user side constitutes a paradigm-shift that avoids a limited vocabulary of query keywords but it also uncovers a series of limitations. By looking at the user needs one can see how current algorithms cannot provide a single general and unifying solution of semantic-information processing. Part of this limitation resides on the problem setting definition as shall be detailed in Section 8.3.

Fortunately, the Web has not only created a vast amount of information but it has also enabled applications for users to communicate and interact. These new ways of communicating define a set of new sources of information (e.g., new types of data, user interaction, or information usage) from where pioneering algorithms can be developed. These new sources of information define an innovative problem setting as we shall discuss in Section 8.4.

Finally, both search by keyword and search by semantic example contribute to friendlier human computer interfaces. However, by pushing the use of semantics across the entire information chain one creates new uses of the studied technology between the two extremes. For example, semantics

can also be used by intermediary systems (e.g., proxies or gateways) to adapt content on-the-fly to user devices with limited capabilities, e.g., mobile phones. We will conclude this chapter with a discussion of the influence of the proposed research on multimedia applications in Section 8.5.

8.2 Future Work

Specific future work concerning the studied algorithms was discussed at the end of the corresponding chapter. Here, I summarize the most important topics:

- **Explore other features** (SIFT, text relations, etc): we limited the set of features to very simple ones as our focus was on the models and not on the features. However, it would be interesting to evaluate the usefulness of more semantic features such as WordNet or other visual grammars.
- **L_1 regularized logistic regression**: sparse models are known to perform better than the smoothed version of logistic regression, e.g., relevance vector machines or support vector machines. This would allow us to use arbitrary dimensions of the feature space and discard the ones that are not in use, thus, reducing the computational complexity. L_1 regularized logistic regression seem ideally suited and has been applied successfully in many fields, see (Park and Hastie 2007).
- **Replace cross-validation**: cross-validation based model selection is computationally very complex and demands large corresponding resources. Other methods for linear models exist that can reduce the model selection cost such as the newly proposed method to follow regularization paths (Park and Hastie 2007).
- **Rich-queries with arbitrary combinations of keywords and semantic-examples**: The presented experiments were all automated retrieval. Users can provide more information (e.g., keywords) to guide the retrieval process towards the correct search topic. Thus, research of rich-query usage would show us new methods of embedding semantics in human computer interfaces.
- **Graph-based semantic dissimilarity**: There is a gap between high-level concepts and high-level search topics. Not all search topics can be represented by those concepts. These observations support the hypothesis that a search topic is not arranged in a sphere type pattern but with some other pattern, e.g., a disjoint mixture of clusters. Graph-based functions should be able to embed concept dependencies in the computation of semantic

dissimilarity.

8.3 Limitations of Semantic-Multimedia IR

Current approaches of extracting information from semantic-multimedia content are based on a scenario that involves several processing steps with an associated loss of information. In this setting the most relevant points of information loss are:

- **Data and annotations:** the traditional semantic-multimedia IR setting is defined by some training data and the corresponding annotations. This is a simple and traditional model that has been explored in different areas and the lack of training data has many times been pointed out as the overriding issue. However, simple annotations are an oversimplification of the problem domain: information semantics are more complex than simple binary annotations of examples. The large amount of weakly-labelled data available from different sources, e.g., Wikipedia, Flickr, or news sites, limits the usability of the traditional setting.
- **Keyword vocabulary:** deploying larger (uncontrolled) vocabularies enabling more expressive descriptions of information has been adopted by several social-media applications. However, as noted by Lew, Sebe et al. (2006) the question “*Is this approach really working – or can it be made to work?*” still remains. I believe that *folksonomies* will suffer the same limitation as Web pages: it is just text and semantics are not associated to the text. This is a consequence of *folksonomy* ambiguity (keywords with no defined meaning) and complexity (keywords interaction are just based in co-occurrence).
- **Low-level representations of information:** computer vision, audio processing and natural language processing all apply a compressed representation of information to enable information extraction algorithms. In all cases, there is a significant loss in the semantics preserved in these low-level representations of data. Thus, richer low-level features that preserve the most relevant information are required.
- **Machine learning:** techniques that allow computers to learn a given task have not yet achieved the level of human understanding and perception required by semantic-multimedia IR. Both similarity functions and learning algorithms are tied to the low-level representation of information, thus suffering the same type of limitations.

As can be inferred from the above list, semantic-multimedia information retrieval combines many different types of expertise and with that it also inherits the different limitations of each area.

8.4 New Challenges in Semantic-Multimedia IR

In this section we discuss new challenges as a set of research topics addressing the limitations listed in the previous section.

Users and Social-Media

In recent years multimedia IR research has focused on the ambitious task of extracting meaningful information from semantic multimedia. Large efforts were allocated to this task, and little effort has been dedicated to the understanding of real user problems (Forsyth 2001). These concerns have again been voiced recently at a talk at the British Computer Society by van Rijsbergen (2007).

Fully automatic analysis algorithms have been pursued feverously by the community. However, a number of social-media applications have successfully inserted the user in the loop, giving evidence that semi-automatic methods are adequate in these scenarios. Social-media applications like Flickr, YouTube⁷, del.icio.us⁸, IMDB⁹, Wikipedia and others have strategically put users in the information processing loop where they are constantly providing valuable feedback.

This new setting contrasts greatly with the classic multimedia IR model and motivates users to cooperate as a large community. Understanding the possibilities of the new problem setting allows scientists to work on solutions that can help users and bring more success to the area of semantic-multimedia IR.

Large-Scale Data Resources

While the traditional problem setting favours supervised methods that model labelled data, the new problem setting makes available large amounts of unlabeled data that create a demand for a new breed of unsupervised algorithms. The objective of these algorithms should be the deduction of a knowledge base concerning the way users perceive and interact with semantic-multimedia information.

Weakly-Labelled Data Resources

Another challenge derives from the community aspect of the new setting. A problem that we found in this thesis is the strong dependence that today's algorithms have on the quality of annotations. Multimedia is currently available with different comments, tags, links and other information that multiple users assign to a given document. This community effect provides algorithms with many implicit (e.g., comments) and explicit (e.g., tags) relevance judgments that

⁷ <http://www.youtube.com>

⁸ <http://del.icio.us/>

⁹ Internet Movie Database: <http://www.imdb.com>

should be exploited.

Both weakly-labelled data resources and the output of unsupervised algorithms are important sources of explicit and implicit annotations of training data. Semi-supervised methods become an attractive type of approach that should be further researched in semantic-multimedia information retrieval.

Cross-Media Information

The new multimedia IR scenarios combine many different types of information sources with different semantics. In this thesis we considered only text and visual information, but many other information sources are available in multimedia, (e.g., authorship, location, event), capturing device characteristics (e.g., lenses depth of field). New algorithms must cope with the multitude of information sources and with the increased complexity and heterogeneity that they exhibit.

8.5 Influence on Multimedia Applications

The research presented in this thesis can make applications aware of semantic-multimedia information that was previously ignored. Besides enabling an all-new set of applications, semantics can also improve existing multimedia applications by:

- Making available richer information
- Improving human computer interaction
- Reducing the application's operational and maintenance costs

These improvements are of great value for multimedia content owners, keepers, distributors and users of multimedia collections. Companies with multimedia product catalogues, TV stations, newspapers, museums, art galleries, picture libraries are currently drowning in non-indexed multimedia information and are eager for new commercial uses of their multimedia assets. All application domains making use of multimedia will benefit from a semantic interaction, for example:

- Architecture, real estate, and interior design, e.g., searching for ideas
- Broadcast media selection, e.g., radio channel, TV channel
- Cultural services, e.g., history museums, art galleries

- Digital libraries, e.g., image catalogue, musical dictionary, bio-medical imaging catalogues, film, video and radio archives
- E-Commerce, e.g., personalized advertising, on-line catalogues, directories of e-shops
- Education, e.g., repositories of multimedia courses, multimedia search for support material
- Entertainment, e.g., systems for the management of personal multimedia collections, including manipulation of content, e.g., home video editing, searching a game, karaoke
- Investigation services, e.g., human characteristics recognition, forensics
- Journalism, e.g., searching speeches of a certain politician using his name, his voice or his face
- Multimedia directory services, e.g., yellow pages, Tourist information, Geographical information systems
- Remote sensing, e.g., cartography, ecology, natural resources management

This list illustrates how our initial objective is fulfilled: the semantic richness of multimedia content enables retrieval applications to deliver more meaningful information.

Nomenclature

Given collection \mathcal{D} of N multimedia documents

$$\mathcal{D} = \{d^1, d^2, \dots, d^N\}, \quad (4.1)$$

each document is characterized by a vector

$$d^j = (d_T^j, d_V^j, d_W^j), \quad (4.2)$$

composed by a feature vector d_T^j describing the text part of document, a feature vector d_V^j describing the visual part of the document, and a keyword vector d_W^j describing the semantics of the document.

To describe the semantics of multimedia information we define a vocabulary

$$\mathcal{W} = \{w_1, \dots, w_L\}, \quad (4.3)$$

of L keywords. The feature vector d_W^j is formally defined as

$$d_W^j = (d_{W,1}^j, d_{W,2}^j, \dots, d_{W,L}^j) \quad (4.4)$$

where each $d_{W,t}^j$ dimension indicates the confidence that keyword w_t is present in document d^j and can be computed automatically by an algorithm

$$d_W^j = p_A(d_T^j, d_V^j), \quad (7.5)$$

or manually assigned by an annotator

$$p_U : d^j \rightarrow d_W^j. \quad (7.6)$$

A Multi-Modal Feature Space

The automatic method is computed as the probability function

$$d_{W,t}^j = p\left(y_t^j = 1 \mid \mathbf{F}\left(d_T^j, d_V^j\right), \beta_t\right). \quad (4.9)$$

where the random variable $y_t^j = \{1, 0\}$ indicates the presence/not-presence of keyword w_t on document d^j and β_t is the keyword model, and the function \mathbf{F} is the optimal feature space transformation

$$\mathbf{F}\left(d_T^j, d_V^j\right) = \left(\mathbf{F}_T\left(d_T^j\right), \mathbf{F}_V\left(d_V^j\right)\right), \quad (4.8)$$

where $\mathbf{F}_T\left(d_T\right)$ is a sparse space transformation and $\mathbf{F}_V\left(d_V\right)$ is a dense space transformation.

The dimensionality of both transformations is selected by the minimum description length criterion,

$$\text{MDL}(msg) = \text{DL}(msg \mid cbk_{\min}) + \text{DL}(cbk_{\min}), \quad (4.13)$$

where cbk_{\min} is the optimal codebook that allows the message msg to be transmitted with the minimum number of bits.

The dense space transformation is defined as the vector

$$\mathbf{F}_{V,j}\left(d_V\right) = \begin{bmatrix} \alpha_{1,j} p\left(d_V \mid \mu_{1,j}, \sigma_{1,j}^2\right) \\ \vdots \\ \alpha_{k_{V,j},j} p\left(d_V \mid \mu_{k_{V,j},j}, \sigma_{k_{V,j},j}^2\right) \end{bmatrix}^T, \quad k_{V,j} \gg n, \quad (4.21)$$

where each dimension is a component of the Gaussian mixture model of the j^{th} low-level visual feature.

The sparse space transformation is defined as the vector

$$\mathbf{F}_T\left(d_{T,1}, \dots, d_{T,n}\right) = \begin{bmatrix} \mathbf{f}_{T,1}\left(d_{T,1}, \dots, d_{T,n}\right) \\ \vdots \\ \mathbf{f}_{T,k_T}\left(d_{T,1}, \dots, d_{T,n}\right) \end{bmatrix}^T, \quad k_T \ll n, \quad (4.23)$$

where each dimension corresponds to a term frequency scaled by its inverted document frequency:

$$f_{T,i}(d_T) = -d_{T,r(i)} \cdot \log \left(\frac{N}{\text{DF}(d_{T,r(i)})} \right), \quad (4.28)$$

where $r(i)$ is a permutation function that returns the i^{th} text term of the information gain rank.

The information gain criteria is expressed as

$$\text{IG}(d_{T,i}) = \frac{1}{L} \sum_{j=1}^L \text{MU}(y_j, d_{T,i}), \quad (4.24)$$

where $d_{T,i}$ corresponds to a text term, y_j indicates the presence of keyword w_j , and MU is the mutual information between these two variables:

$$\text{MU}(y_j, t_i) = \sum_{y_j=\{0;1\}} \sum_{d_{T,i}} p(y_j, d_{T,i}) \log \frac{p(y_j, d_{T,i})}{p(y_j) p(d_{T,i})}, \quad (4.25)$$

Keyword Models

Keywords models β_t in the optimal feature space defined by $F(d)$ assume the following forms:

- A Rocchio classifier with a cosine distance

$$\cos(\beta_t, F(d)) = \frac{\beta_t}{\|\beta_t\|} \cdot \frac{F(d)}{\|F(d)\|}, \quad (5.6)$$

where the keyword model corresponds to the regression coefficient

$$\beta_t = \frac{1}{|\mathcal{D}_{w_t}|} \sum_{d \in \mathcal{D}_{w_t}} \frac{F(d)}{\|F(d)\|} - \frac{1}{|\mathcal{D} \setminus \mathcal{D}_{w_t}|} \sum_{d \in \mathcal{D} \setminus \mathcal{D}_{w_t}} \frac{F(d)}{\|F(d)\|}. \quad (5.5)$$

- A naïve Bayes model:

$$\log \frac{p(y_t^j = 1 | d^j)}{p(y_t^j = 0 | d^j)} = \log \frac{p(y_t^j = 1)}{p(y_t^j = 0)} + M \sum_{i=1}^M p(f_i | d^j) \log \frac{p(f_i | y_t^j = 1)}{p(f_i | y_t^j = 0)} \quad (5.16)$$

- A logistic regression model with the likelihood function defined as:

$$l(\beta_t) = \sum_{d^j \in \mathcal{D}} \left(y_t^j \beta_t F(d^j) - \log(1 + \exp(\beta_t F(d^j))) \right) - \lambda \beta_t^2, \quad (5.27)$$

$$\lambda = \frac{1}{2\sigma_\xi^2},$$

where σ_ξ^2 is the variance of the Gaussian prior, and y_t^j corresponds to the annotation of keyword w_t on document d^j .

Keyword Spaces

A keyword space is defined by the following properties:

- **Vocabulary:** defines a lexicon \mathcal{W} of L keywords used to annotate multimedia documents.
- **Multimedia keyword vectors:** a multimedia document is represented by the vector of keywords d_W .
- **Keyword vectors computation:** the keyword vector can be computed automatically or provided by a user.
- **Semantic dissimilarity:** given a keyword space defined by the vocabulary \mathcal{W} , semantic dissimilarity between two documents is defined as

$$\text{dissim}_w : [0,1]^L \times [0,1]^L \rightarrow \mathcal{R}_0^+, \quad (7.3)$$

the function in the L dimensional space that returns the distance between two keyword vectors.

The following dissimilarity functions were tested:

$$D_{\text{Minkowski}}(q_W, d_W) = L_p(q_W, d_W) = \left[\sum_{i=1}^L |q_{W,i} - d_{W,i}|^p \right]^{1/p}, \quad (7.13)$$

$$D_{\text{Manhattan}}(q_W, d_W) = L_1(q_W, d_W) = \sum_{i=0}^L |q_{W,i} - d_{W,i}|, \quad (7.14)$$

$$D_{\text{Euclidean}}(q_W, d_W) = L_2(q_W, d_W) = \sqrt{\sum_{i=0}^L (q_{W,i} - d_{W,i})^2}, \quad (7.15)$$

$$D_{\text{Chebyshev}}(q_W, d_W) = L_\infty(q_W, d_W) = \max_{0 \leq i \leq L} |q_{W,i} - d_{W,i}|, \quad (7.16)$$

$$D_{\text{Cosine}}(q_W, d_W) = \cos(q_W \angle d_W) = 1 - \frac{q_W \cdot d_W}{\|q_W\| \cdot \|d_W\|}, \quad (7.17)$$

$$D_{\text{Canberra}}(q_W, d_W) = \sum_{i=1}^L \frac{|q_{W,i} - d_{W,i}|}{|q_{W,i}| + |d_{W,i}|}, \quad (7.19)$$

$$D_{\text{KL}}(q_W \parallel d_W) = \sum_{i=1}^L p(q_{W,i}) \log \frac{p(q_{W,i})}{p(d_{W,i})}, \quad (7.20)$$

$$D_{\text{JS}}(q_W, d_W) = \frac{1}{2} D_{\text{KL}} \left(q_W \parallel \frac{1}{2}(q_W + d_W) \right) + \frac{1}{2} D_{\text{KL}} \left(d_W \parallel \frac{1}{2}(q_W + d_W) \right). \quad (7.21)$$

References

- Adams, W. H., Iyengart, G., Lin, C. Y., Naphade, M. R., Neti, C., Nock, H. J., and Smith, J. R. (2003). Semantic indexing of multimedia content using visual, audio and text cues. *EURASIP Journal on Applied Signal Processing* 2003 (2):170-185.
- Aizerman, M., Braverman, E., and Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control* 25:821-837.
- Akutsu, M., Hamada, A., and Tonomura, Y. (1998). Video handling with music and speech detection. *IEEE Multimedia* 5 (3):17-25.
- Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM* 26 (11):832-843.
- Aslam, J. A., and Yilmaz, E. (2007). Inferring document relevance from incomplete information. In *ACM Conf. on information and knowledge management*, November 2007, Lisbon, Portugal.
- Bach, J. R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R. C., and Shu, C.-F. (1996). Virage image search engine: an open framework for image management. In *Proc. SPIE Int. Soc. Opt. Eng.*, San Jose.
- Baeza-Yates, R., and Ribeiro-Neto, B. (1999). *Modern information retrieval*. New York: Addison Wesley.
- Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D. M., and Jordan, M. I. (2003). Matching words and pictures. *Journal of Machine Learning* 3 (6):1107-1135.
- Barnard, K., and Forsyth, D. A. (2001). Learning the semantics of words and pictures. In *Int'l Conf. on Computer Vision*, 2001, Vancouver, Canada.
- Barron, A., and Cover, T. (1991). Minimum complexity density estimation. *IEEE Transactions on Information Theory* 37 (4):1034-1054.
- Benitez, A. (2005). *Multimedia knowledge: discovery, classification, browsing, and retrieval*. PhD Thesis, Graduate School of Arts and Sciences, Columbia University, New York.
- Benitez, A. B., and Chang, S. F. (2002). Multimedia knowledge integration, summarization and evaluation. In *Int'l Workshop on Multimedia Data Mining in conjunction with the Int'l Conf. on Knowledge Discovery & Data Mining*, July 2002, Alberta, Canada.

- Benitez, A. B., Smith, J. R., and Chang, S.-F. (2000). MediaNet: A Multimedia Information Network for Knowledge Representation. In *SPIE Conference on Internet Multimedia Management Systems* Nov 2000, Boston, MA, USA.
- Berger, A., Pietra, S., and Pietra, V. (1996). A maximum entropy approach to natural language processing. In *Computational Linguistics*, 1996.
- Blei, D., and Jordan, M. (2003). Modeling annotated data. In *ACM SIGIR Conf. on research and development in information retrieval*, August 2003, Toronto, Canada.
- Buckley, C., and Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *ACM SIGIR Conf. on research and development in information retrieval*, August 2000, Athens, Greece.
- . (2004). Retrieval evaluation with incomplete information. In *ACM SIGIR Conf. on research and development in information retrieval*, July 2004, Sheffield, United Kingdom.
- Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. (2003). Extracting content structure for Web pages based on visual representation. In *Asia Pacific Web Conference 2003*.
- Carneiro, G., and Vasconcelos, N. (2005). Formulating semantic image annotation as a supervised learning problem. In *IEEE Conf. on Computer Vision and Pattern Recognition*, August 2005, San Diego, CA, USA.
- Chang, S.-F., Hsu, W., Kennedy, L., Xie, L., Yanagawa, A., Zavesky, E., and Zhang, D.-Q. (2005). Columbia University TRECVID-2005 video search and high-level feature extraction. In *TRECVID*, November 2005, Gaithersburg, MD.
- Chen, S. F., and Rosenfeld, R. (1999). A Gaussian prior for smoothing maximum entropy models. Technical Report, Carnegie Mellon University, Pittsburg, PA, February 1999.
- Cover, T. M., and Thomas, J. A. (1991). *Elements of information theory*, *Wiley Series in Telecommunications*: John Wiley & Sons.
- Cox, I. J., Miller, M. L., Omohundro, S. M., and Yianilos, P. N. (1996). PicHunter: Bayesian relevance feedback for image retrieval. In *Proceedings of the International Conference on Pattern Recognition*.
- Croft, W. B., Turtle, H. R., and Lewis, D. D. (1991). The use of phrases and structured queries in information retrieval. In *ACM SIGIR Conf. on research and development in information retrieval*, Chicago, Illinois, United States.
- Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: ideas, influences, and trends of the new age. *ACM Computing Surveys*.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41 (6):391-407.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern classification*. John Wiley & Sons.
- Duygulu, P., Barnard, K., de Freitas, N., and Forsyth, D. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European Conf. on Computer Vision*, May 2002, Copenhagen, Denmark.

- Ekin, A., Tekalp, A. M., and Mehrotra, R. (2003). Automatic video analysis and summarization. *IEEE Transactions on Image Processing* 12 (7):796-807.
- Feng, S. L., Lavrenko, V., and Manmatha, R. (2004). Multiple Bernoulli relevance models for image and video annotation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2004, Cambridge, UK.
- Figueiredo, M., and Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (3):381-396.
- 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. (1995). Query by image and video content: the QBIC system. *IEEE Computer* 28 (9):23-32.
- Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Machine Learning Research*:1289-1305.
- Forsyth, D. (2001). Benchmarks for storage and retrieval in multimedia databases. Technical Report, Computer Science Division, U.C. Berkeley, Berkeley.
- Forsyth, D., and Ponce, J. (2003). *Computer vision: a modern approach*. Prentice Hall.
- Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian data analysis*. 2nd ed: Chapman & Hall / CRC.
- Grubinger, M., Clough, P., HanburyAllan, and Müller, H. (2007). Overview of the ImageCLEF 2007 Photographic Retrieval Task. In *Working Notes of the 2007 CLEF Workshop*, September 2007, Budapest, Hungary.
- Harabagiu, S., Moldovan, D., Pasaca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V., and Morarescu, P. (2000). FALCON: Boosting knowledge for answer engines. In *Text REtrieval Conf.*, November 2000, Gaithersburg, MD, USA.
- Hare, J., Samangoeei, S., and Lewis, P. H. (2008). Semantic spaces revisited: investigating the performance of auto-annotation and semantic retrieval using semantic spaces. In *ACM Conf. on image and video retrieval*, July 2008, Niagara Falls, Canada.
- Hare, J. S., Lewis, P. H., Enser, P. G. B., and Sandom, C. J. (2006). A linear-algebraic technique with an application in semantic image retrieval. In *Int'l Conference on Image and Video Retrieval*, July 2006, Phoenix, AZ, USA.
- Hartley, R., and Zisserman, A. (2004). *Multiple view geometry in computer vision* 2nd ed: Cambridge University Press.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The elements of statistical learning: Data mining, inference and prediction*, Springer Series in Statistics: Springer.
- Haubold, A., Natsev, A., and Naphade, M. (2006). Semantic multimedia retrieval using lexical query expansion and model-based re-ranking. In *IEEE Int'l Conference on Multimedia and Expo*, July 2006, Toronto, Canada.

- Hauptmann, A., Yan, R., and Lin, W.-H. (2007). How many high-level concepts will fill the semantic gap in news video retrieval? In *ACM Conf. on image and video retrieval*, July 2007, Amsterdam, The Netherlands.
- He, X., King, O., Ma, W.-Y., Li, M., and Zhang, H.-J. (2003). Learning a semantic space from user's relevance feedback for image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology* 13 (1):39- 48.
- He, X., Zemel, R. S., and Carreira-Perpiñán, M. Á. (2004). Multiscale conditional random fields for image labeling. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, June 2004, Cambridge, UK.
- Heesch, D. (2005). *The NNk technique for image searching and browsing*. PhD Thesis, Department of Computing, University of London, Imperial College of Science, Technology and Medicine, London, UK.
- Heesch, D., and Rüger, S. (2004). Three interfaces for content-based access to image collections. In *Int'l Conf. on Image and Video Retrieval*, July 2004, Dublin, Ireland.
- Hofmann, T. (1999). Probabilistic latent semantic indexing. In *ACM SIGIR Conf. on research and development in information retrieval*, August 1999, Berkeley, CA, USA.
- Hofmann, T., and Puzicha, J. (1998). Statistical models for co-occurrence data. Technical Report, Massachusetts Institute of Technology, 1998.
- Howarth, P. (2007). *Discovering images: features, similarities and subspaces*. PhD Thesis, Department of Computing, University of London, Imperial College of Science, Technology and Medicine, London.
- Howarth, P., and Rüger, S. (2005a). Fractional distance measures for content-based image retrieval. In *European Conference on Information Retrieval*, April 2005, Santiago de Compostela, Spain.
- . (2005b). Trading accuracy for speed. In *Int'l Conf. on Image and Video Retrieval*, July 2005, Singapore.
- Huijsmans, D. P., and Sebe, N. (2005). How to complete performance graphs in content-based image retrieval: add generality and normalize scope. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2):245-251.
- iProspect. Access (April 2006). *Search engine user behavior study* April 2006. Available from http://www.iprospect.com/about/whitepaper_seuserbehavior_apr06.htm.
- Jain, R. (2001). Knowledge and experience. *IEEE Multimedia* 8 (4):4.
- Jeon, J., Lavrenko, V., and Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR Conf. on research and development in information retrieval*, September 2003, Toronto, Canada.
- Jeon, J., and Manmatha, R. (2004). Using maximum entropy for automatic image annotation. In *Int'l Conf on Image and Video Retrieval*, July 2004, Dublin, Ireland.

- Jiang, Y.-G., Ngo, C.-W., and Yang, J. (2007). Towards optimal bag-of-features for object categorization and semantic video retrieval. In *ACM Conf. on image and video retrieval*, July 2007, Amsterdam, The Netherlands.
- Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Int'l Conf. on Machine Learning*, July 1997, Nashville, US.
- . (1998). Text categorization with Support Vector Machines: learning with many relevant features. In *European Conf. on Machine Learning*, September 1998.
- Jose, J. M., Furner, J. F., and Harper, D. J. (1998). Spatial querying for image retrieval: a user-oriented evaluation. In *ACM SIGIR Conference on research and development in information retrieval*, August 1998, Melbourne, Australia.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint Conference on artificial intelligence*, August 1995, Montréal, Québec, Canada.
- Kokare, M., Chatterji, B. N., and Biswas, P. K. (2003). Comparison of similarity metrics for texture image retrieval. In *IEEE TENCON 2003*, Oct. 2003, Bangalore, India.
- Kumar, S., and Herbert, M. (2003a). Discriminative random fields: A discriminative framework for contextual interaction in classification. In *IEEE Int'l Conf. on Computer Vision*, October 2003, Nice, France.
- . (2003b). Man-made structure detection in natural images using causal multiscale random field. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, June 2003, Madison, WI, USA.
- Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Int'l Conf. on Machine Learning*, June 2001, San Francisco, CA, USA.
- Lavrenko, V., Manmatha, R., and Jeon, J. (2003). A model for learning the semantics of pictures. In *Neural Information Processing System Conf.*, December 2003, Vancouver, Canada.
- Leonardi, R., Miglotari, P., and Prandini, M. (2004). Semantic indexing of soccer audio-visual sequences: A multimodal approach based on controlled Markov chains. *IEEE Transactions on Circuits Systems and Video Technology* 14 (5):634-643.
- Lew, M. S., Sebe, N., Djeraba, C., and Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications* 2 (1):1-19.
- Li, B., and Sezan, I. (2003). Semantic sports video analysis: approaches and new applications. In *IEEE Int'l Conf. on Image Processing*, September 2003, Barcelona, Spain.
- Li, D., Dimitrova, N., Li, M., and Sethi, I. (2003). Multimedia content processing through cross-modal association. In *ACM Conf. on Multimedia*, November 2003, Berkeley, California, USA.

- 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 (9):1075-1088.
- Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Trans. Inform. Theory* 37 (1):145-151.
- Liu, D. C., and Nocedal, J. (1989a). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*.
- . (1989b). On the limited memory method for large scale optimization. *Mathematical Programming B* 45 (3):503-528.
- Lowe, D. (1999). Object recognition from local scale-invariant features. In *Int. Conf. on Computer Vision*, September 1999, Kerkyra, Corfu, Greece.
- Lu, L., Zhang, H.-J., and Jiang, H. (2002). Content analysis for audio classification and segmentation. *IEEE Transactions on Speech and Audio Processing* 10 (7):293-302.
- Lu, Y., Hu, C., Zhu, X., Zhang, H., and Yang, Q. (2000). A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *ACM Conf. on Multimedia*, October 30 - November 3, Los Angeles, CA, USA.
- Luo, Y., and Hwang, J. N. (2003). Video sequence modeling by dynamic Bayesian networks: A systematic approach from coarse-to-fine grains. In *IEEE Int'l Conf. on Image Processing*, September 2003, Barcelona, Spain.
- MacKay, D. J. C. (2004). *Information theory, inference, and learning algorithms*. Cambridge: Cambridge University Press.
- Magalhães, J., Overell, S., and Rüger, S. (2007). A semantic vector space for query by image example. In *ACM SIGIR Conf. on research and development in information retrieval, Multimedia Information Retrieval Workshop*, July 2007, Amsterdam, The Netherlands.
- Magalhães, J., and Pereira, F. (2004). Using MPEG standards for multimedia customization. *Signal Processing: Image Communication* 19 (5):437-456.
- Magalhães, J., and Rüger, S. (2006). Semantic multimedia information analysis for retrieval applications. In *Semantic-Based Visual Information Retrieval*, edited by Y.-J. Zhang: IDEA group publishing.
- . (2007a). High-dimensional visual vocabularies for image retrieval. In *ACM SIGIR Conf. on research and development in information retrieval*, July 2007, Amsterdam, The Netherlands.
- . (2007b). Information-theoretic semantic multimedia indexing. In *ACM Conf. on Image and Video Retrieval*, July 2007, Amsterdam, The Netherlands.
- Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. *Sixth Conf. on Natural Language Learning*:49-55.
- Marr, D. (1983). *Vision*. San Francisco: W. H. Freenman.

- McCallum, A., and Nigam, K. (1998). A comparison of event models for naive Bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, 1998.
- McCullagh, P., and Nelder, J. A. (1989). *Generalized linear models*. 2nd ed: Chapman and Hall.
- Miller, G. A. (1995). WORDNET: A lexical database for English. *Communications of ACM* 38 (11):39-41.
- Mizzaro, S. (1997). Relevance: the whole history. *Journal of the American Society of Information Science* 48 (9):810-832.
- Monay, F., and Gatica-Perez, D. (2007). Modeling semantic aspects for cross-media image indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (10):1802-1817.
- Mori, Y., Takahashi, H., and Oka, R. (1999). Image-to-word transformation based on dividing and vector quantizing images with words. In *First Int'l Workshop on Multimedia Intelligent Storage and Retrieval Management*, October 1999, Orlando, FL, USA.
- Müller, H., Marchand-Maillet, S., and Pun, T. (2002). The truth about Corel - Evaluation in image retrieval. In *Int'l Conf. on Image and Video Retrieval*, July 2002, London, UK.
- Murphy, K., Torralba, A., and Freeman, W. T. (2003). Using the forest to see the trees: A graphical model relating features, objects and scenes. In *Neural Information Processing Systems Conf.*, December 2003, Vancouver, Canada.
- Naphade, M., Mehrotra, R., Ferman, A. M., Warnick, J., Huang, T. S., and Tekalp, A. M. (1998). A high performance shot boundary detection algorithm using multiple cues. In *IEEE Int'l Conf. on Image Processing*, October 1998, Chicago, IL, USA.
- Naphade, M., and Smith, J. (2003). Learning visual models of semantic concepts. In *IEEE Int'l Conf. on Image Processing*, September 2003, Barcelona, Spain.
- Naphade, M., Smith, J. R., Tesic, J., Chang, S.-F., Hsu, W., Kennedy, L., Hauptmann, A., and Curtis, J. (2006). Large-scale concept ontology for multimedia. *IEEE Multimedia Magazine* 13 (3):86-91.
- Naphade, M. R., and Huang, T. S. (2000). Stochastic modeling of soundtrack for efficient segmentation and indexing of video. In *SPIE, Storage and Retrieval for Media Databases*, January 2000, San Jose, CA, USA.
- . (2001). A probabilistic framework for semantic video indexing filtering and retrieval. *IEEE Transactions on Multimedia* 3 (1):141-151.
- Natsev, A., Haubold, A., Tesic, J., Xie, L., and Yan, R. (2007). Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *ACM Conf. on Multimedia*, September 2007, Augsburg, Germany.
- Natsev, A., Naphade, M., and Smith, J. (2003). Exploring semantic dependencies for scalable concept detection. In *IEEE Int'l Conf. on Image Processing*, September 2003, Barcelona, Spain.

- Nigam, K., Lafferty, J., and McCallum, A. (1999). Using maximum entropy for text classification. In *IJCAI - Workshop on Machine Learning for Information Filtering*, August 1999, Stockholm, Sweden.
- Nocedal, J., and Wright, S. J. (1999). *Numerical optimization*. New York: Springer-Verlag.
- Ortega, M., Rui, Y., Chakrabarti, K., Mehrotra, S., and Huang, T. S. (1997). Supporting similarity queries in MARS. In *ACM Conf. on Multimedia*, Seattle, Washington, United States.
- Park, M.-Y., and Hastie, T. (2007). An L1 regularization-path algorithm for generalized linear models. *Journal of the Royal Statistics Society B* 69 (4):659-677.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Los Angeles: Morgan Kaufmann Publishers.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program* 14 (3):130-137.
- Quattoni, A., Collins, M., and Darrell, T. (2004). Conditional random fields for object recognition. In *Neural Information Processing Systems Conf.*, December 2004, Vancouver, Canada.
- . (2007). Learning visual representations using images with captions. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, Minneapolis, MN, USA.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE* 77 (2):257-286.
- Rasiwasia, N., Moreno, P., and Vasconcelos, N. (2007). Bridging the gap: Query by semantic example. *IEEE Transactions on Multimedia* 9 (5):923-938.
- Rasiwasia, N., Vasconcelos, N., and Moreno, P. (2006). Query by semantic example. In *CIVR*, July 2006, Phoenix, AZ, USA.
- Rijsbergen, C. J. v. (2007). *BCS 50th anniversary talk: Past, present and future of Information Retrieval*. London, 22 May 2007.
- Rissanen, J. (1978). Modeling by shortest data description. *Automatica* 14:465-471.
- Rocchio, J. (1971). Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Text Retrieval*, edited by G. Salton: Prentice-Hall.
- Rui, Y., Huang, T., Ortega, M., and Mehrotra, S. (1998). Relevance feedback: a power toll for interactive content-based image retrieval. *IEEE Transactions on Circuits Systems and Video Technology* 8 (5):644-655.
- Sha, F., and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Human Language Technology Conf. of the North American Chapter of the Association for Computational Linguistics*, May 2003, Edmonton, Canada.
- Shi, J., and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8):888-905.

- Smeaton, A. F., and Quigley, I. (1996). Experiments on using semantic distances between words in image caption retrieval. In *ACM SIGIR Conf. on research and development in information retrieval*, July 1996, Zurich, Switzerland.
- 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., and Chang, S.-F. (1996). VisualSEEk: a fully automated content-based image query system. In *ACM Conf. on Multimedia*, November 1996, Boston, MA, USA.
- Snoek, C. G. M., and Worring, M. (2005a). Multimedia event based video indexing using time intervals. *IEEE Transactions on Multimedia* 7 (4).
- . (2005b). Multimodal video indexing: a review of the state-of-the-art. *Multimedia Tools and Applications* 25 (1):5-35.
- Snoek, C. G. M., Worring, M., Geusebroek, J.-M., Koelma, D. C., Seinstra, F. J., and Smeulders, A. W. M. (2006). The semantic pathfinder: using an authoring metaphor for generic multimedia indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (10):1678-1689.
- Souvannavong, F., Merialdo, B., and Huet, B. (2003). Latent semantic indexing for video content modeling and analysis. In *TREC Video Retrieval Evaluation Workshop*, November 2003, Gaithersburg, MD, USA.
- Srikanth, M., Varner, J., Bowden, M., and Moldovan, D. (2005). Exploiting ontologies for automatic image annotation. In *ACM SIGIR Conf. on research and development in information retrieval*, August 2005, Salvador, Brazil.
- Sundaram, H., and Chang, S. F. (2000). Determining computable scenes in films and their structures using audio visual memory models. In *ACM Conf. on Multimedia*, October 2000, Los Angeles, CA, USA.
- Swain, M. J., and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision* 7 (1):11-32.
- Tan, Y.-P., Saur, D. D., Kulkarni, S. R., and Ramadge, P. J. (2000). Rapid estimation of camera motion from compressed video with application to video annotation. *IEEE Transactions on Circuits and Systems for Video Technology* 10 (1):133-146.
- Tansley, R. (2000). *The multimedia thesaurus: Adding a semantic layer to multimedia information*. PhD Thesis, University of Southampton, Southampton, UK.
- Tesic, J., Natsev, A., and Smith, J. R. (2007). Cluster-based data modelling for semantic video search. In *ACM Conf. on Image and Video Retrieval*, July 2007, Amsterdam, The Netherlands.
- Torralba, A., Murphy, K., and Freeman, W. (2004). Contextual models for object detection using boosted random fields. In *Neural Information Processing Systems Conf.*, December 2004, Vancouver, Canada.

- Town, C. P., and Sinclair, D. A. (2004). Language-based querying of image collections on the basis of an extensible ontology. *International Journal of Image and Vision Computing* 22 (3):251-267.
- Tseng, B. L., Lin, C.-Y., Naphade, M., Natsev, A., and Smith, J. (2003). Normalised classifier fusion for semantic visual concept detection. In *IEEE Int'l Conf. on Image Processing*, September 2003, Barcelona, Spain.
- Turtle, H., and Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. *ACM Trans. Inf. Syst* 9 (3):187-222.
- Tzanetakis, G., and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10 (5):293-302.
- Urban, J., Jose, J. M., and Rijsbergen, C. J. V. (2003). An adaptive approach towards content-based image retrieval. In *Workshop on content based multimedia indexing*, September 2003, Rennes, France.
- Vailaya, A., Figueiredo, M., Jain, A., and Zhang, H. (1999). A Bayesian framework for semantic classification of outdoor vacation images. In *SPIE: Storage and Retrieval for Image and Video Databases VII*, January, 1999, San Jose, CA, USA.
- Vailaya, A., Figueiredo, M., Jain, A. K., and Zhang, H. J. (2001). Image classification for content-based indexing. *IEEE Transactions on Image Processing* 10 (1):117-130.
- Vasconcelos, N. (2000). *Bayesian models for visual information retrieval*. PhD Thesis, MIT, Cambridge, MA, USA.
- . (2004). On the efficient evaluation of probabilistic similarity functions for image retrieval. *IEEE Transactions on Information Theory* 50 (7):1482-1496.
- Vasconcelos, N., and Lippman, A. (1998). A Bayesian framework for semantic content characterization. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 1998, Santa Barbara, CA, USA.
- . (2000). Statistical models of video structure for content analysis and characterization. *IEEE Transactions on Image Processing* 9 (1):1-17.
- Volkmer, T., Thom, J. A., and Tahaghoghi, S. M. M. (2007). Modeling human judgment of digital imagery for multimedia retrieval. *IEEE Transactions on Multimedia* 9 (7):967-974.
- Voorhees, E. M. (1998). Variations in relevance judgments and the measurement of retrieval effectiveness. In *ACM SIGIR Conf. on research and development in information retrieval*, August 1998, Melbourne, Australia.
- . (2001). Evaluation by highly relevant documents. In *ACM SIGIR Conf. on Research and development in information retrieval*, July 2001, New Orleans, Louisiana, United States.
- Wactlar, H. D., Kanade, T., Smith, M. A., and Stevens, S. M. (1996). Intelligent access to digital video: Informedia project. *IEEE Computer* 29 (5):46-52.

- Wang, J. Z., Li, J., and Wiederhold, G. (2001). SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (9):947-963.
- Westerveld, T., and de Vries, A. P. (2003a). Experimental evaluation of a generative probabilistic image retrieval model on 'easy' data. In *Multimedia Information Retrieval Workshop in conjunction with ACM SIGIR Conf. on research and development in information retrieval*, July 2003, Toronto, Canada.
- . (2003b). Experimental result analysis for a generative probabilistic image retrieval model. In *ACM SIGIR Conf. on research and development in information retrieval*, July 2003, Toronto, Canada.
- Westerveld, T., de Vries, A. P., Ianeva, T., Boldareva, L., and Hiemstra, D. (2003). Combining information sources for video retrieval. In *TREC Video Retrieval Evaluation Workshop*, November 2003, Gaithersburg, MD, USA.
- Wu, Y., Chang, E., Chang, K., and Smith, J. (2004). Optimal multimodal fusion for multimedia data analysis. In *ACM Conf. on Multimedia*, October 2004, New York, USA.
- Yang, C., Dong, M., and Fotouhi, F. (2005). Semantic feedback for interactive image retrieval. In *Int'l Multimedia Modelling Conference*, January 2005, Singapore.
- Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*:69-90.
- Yang, Y., and Chute, C. G. (1994). An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems* 13 (3):252-277.
- Yang, Y., and Liu, X. (1999). A re-examination of text categorization methods. In *SIGIR*, August 1999.
- Yang, Y., and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Int'l Conf. on Machine Learning*, July 1997, Nashville, Tennessee, USA.
- Yavlinsky, A. (2007). *Image indexing and retrieval using automated annotation*. PhD Thesis, Department of Computing, University of London, Imperial College of Science, Technology and Medicine, London.
- Yavlinsky, A., and Rüger, S. (2007). Efficient re-indexing of automatically annotated image collections using keyword combination. In *SPIE*, January 2007, San Jose, CA, USA.
- Yavlinsky, A., Schofield, E., and Rüger, S. (2005). Automated image annotation using global features and robust nonparametric density estimation. In *Int'l Conf. on Image and Video Retrieval*, July 2005, Singapore.
- Yilmaz, E., and Aslam, J. A. (2006). Estimating average precision with incomplete and imperfect judgments. In *ACM Conf. on information and knowledge management*, November 2006, Arlington, Virginia, USA.
- Yu, J., Amores, J., Sebe, N., Radeva, P., and Tian, Q. (2008). Distance learning for similarity estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (3):451-462.

- Yu, S., Cai, D., Wen, J.-R., and Ma, W.-Y. (2003). Improving pseudo-relevance feedback in Web Information Retrieval using Web page segmentation. In *Int'l World Wide Web Conference*, May 2003, Budapest, Hungary.
- Zhang, C., and Chen, T. (2002). An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia* 4 (2):260- 268.
- Zhang, T., and Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval*:5-31.
- Zhao, R., and Grosky, W. I. (2003). Negotiating the semantic gap: from feature maps to semantic landscapes. *Pattern Recognition* 35 (3):593-600.
- Zheng, Y.-T., Neo, S.-Y., Chua, T.-S., and Tian, Q. (2008). Probabilistic optimized ranking for multimedia semantic concept detection via RVM. In *ACM Conf. on image and video retrieval* July 2008, Niagara Falls, Canada.