Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval

Rong Yan

CMU-LTI-06-008

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA, 15213
www.lti.cs.cmu.edu

Thesis Committee:
Alexander G. Hauptmann (chair)
Christos Faloutsos
John Lafferty
John R. Smith, IBM TJ Waston Research Center

Submitted in partial fulfillment of the requirements
of the degree of Doctor of Philosophy
In Language and Information Technologies.

Abstract

In recent years, the multimedia retrieval community is gradually shifting its emphasis from analyzing one media source at a time to exploring the opportunities of combining diverse knowledge sources from correlated media types and context. In order to combine multimedia knowledge sources, two basic issues must be addressed: *what* to combine and *how* to combine. While considerable effort has been expended to generate a wide range of ranking features from knowledge sources, relatively less attention has been given to the problem of finding a suitable strategy to combine them. It has always been a significant challenge to develop principled combination approaches and capture useful factors such as query information and context information in the retrieval process.

This thesis presents a conditional probabilistic retrieval model as a principled framework to combine diverse knowledge sources. This model can integrate multiple forms of ranking features (query dependent and query independent features) as well as query information and context information in a unified framework with a solid probabilistic foundation. Under this retrieval framework, we overview and develop a number of state-of-the-art approaches for extracting ranking features from multimedia knowledge sources. In order to deal with heterogenous features, a discriminative learning approach is suggested for estimating the combination parameters. Moreover, an efficient rank learning approach has been developed to explicitly model the ranking relations in the learning process with much less training time.

To incorporate query information in the combination model, this thesis develops a number of *query analysis* models that can automatically discover mixing structure of the query space based on previous retrieval results, and predict combination parameters for unseen queries. In more detail, we propose the *query-class based analysis* model which needs to manually define the query classes and a series of *probabilistic latent query analysis*(pLQA) models which can automatically discover latent query classes from the development data by unifying the combination weight optimization and query class categorization into a discriminative learning framework. To adapt the combination function on a per query basis, this thesis also presents a *probabilistic local context analysis*(pLCA) model to automatically leverage additional retrieval sources to improve initial retrieval outputs. A pLCA variant is proposed to utilize human feedback to adjust combination parameters.

All the proposed approaches are evaluated on multimedia retrieval tasks with large-scale video collections. Beyond multimedia collections, we also evaluate our approaches on meta-search tasks with large-scale text collections. Experimental evaluations demonstrate the promising performance of the probabilistic retrieval framework with query analysis and context analysis in the task of knowledge source combination. The applicability of the proposed methods can be extended to many other areas, such as question answering, web IR, cross-lingual IR, multi-sensor fusion, human tracking, and so forth.

To Dear Yan and my parents for love and support.

Acknowledgements

During my studies at Carnegie Mellon University, I have met many people who have provided priceless contributions to this work. I could always rely on their support, sympathy, and wisdom, and they deserve my most tender acknowledgment.

First and foremost, I would like to thank my advisor Alex Hauptmann for his great guidance and timely support over the last five years. I am grateful that he gave me the invaluable opportunity to join the Informedia project when I was making little progress on exploring my PhD research topic in the first year. What I learned from him is far more than I expected, especially in the field of multimedia content analysis. I also appreciate the freedom I was given to work with different collaborators in CMU, which has greatly broadened my understanding across other research areas. It has been a great pleasure to have the opportunity to work with such a true expert in the field.

I would also like to thank my other thesis committee members, Christos Faloutsos, John Lafferty, and John R. Smith, for their invaluable assistance, feedback and patience at all stages of this thesis. Their criticisms, comments, and advice were critical in making this thesis more accurate, more complete and clearer to read.

Moreover, I am also indebted to fellow group members in the Informedia project for providing me inspiration and suggestions during the meetings and the annual TRECVID evaluations. My special thanks go to Jun Yang, Robert Chen, Wei-hao Lin, Cees Snoek, and Pinar Duygulu for their insightful discussions and efforts to generate a large number of semantic concepts that serve as the foundation of this thesis. Jie Yang, Datong Chen, Eric Xing, Jian Zhang and Rong Jin also deserve my special thanks. They are my outstanding collaborators on numerous publications during this PhD tenure.

Many ideas in this thesis started to take shape during an internship at IBM T.J. Waston research center in the summer of 2004, where I had the opportunity to work with my mentor, Milind R. Naphade, and manager, Murray Campbell.

Their great guidance help me tremendously in understanding the area of multimedia information retrieval. I also want to thank my other colleagues at IBM, Paul Natsev, Jelena Tesic, Ching-Yung Lin and Gang Wu, that help me walk through the painful TRECVID evaluation that year.

Good friends are difficult to come by. I owe many thanks to my good friends and fellow graduate students, Jiazhi Ou, Jie Lu, Fan Li, Luo Si, Joy Zhang, Hua Yu, Zhong Xiu, Jerry Zhu, Ke Yang, Yiheng Li, Minglong Shao, Yanjun Qi, Ting Liu, Wen Wu, Hai Huang, Haifeng Zhu, Huan Chang, Juchang Hua, Yan Li, Hong Yan, Zheng Wang, Di Yu, Jiajun Chen, and Qifa Ke for their good friendship and support.

Last, but definitely not the least, I would also like to thank my family for their love and support, without which I would not have survived the Ph.D. process.

Contents

List of Figures

List of Tables

Notation

$\mathcal{Q}$	The set of all queries for retrieval
Q_t	A query with keywords and possibly image/video examples
q_{tl}	A query feature for query Q_t
M_Q	Number of all queries
$\mathcal{D}$	The set of all documents for retrieval
D_j	A multimedia document with features from multi-modalities
M_D	Number of all documents
$f_i(D, Q)$	A ranking feature associated with document D and query Q
λ_i	A combination parameter for ranking features
z	Variable of (latent) query classes
μ	Multinomial parameters for query class variable z
$P(y_+\|D, Q)$	The conditional probability that document D is relevant to query Q
$P(y_-\|D, Q)$	The conditional probability that document D is irrelevant to query Q

Chapter 1

Introduction

The first computer-based use of multimedia data was developed in the early 1960's, which attempted to combine the text and images in a document. Soon thereafter, more and more continuous media, e.g., audio, animations, and video, were incorporated in multimedia systems. Nowadays most people refer to multimedia as the idea of combining different media sources into one application [RJ04], such as broadcast news video that uses text, images, and audio to describe the progress of news events. Another example of multimedia is a distributed collaboration system that enables remote people to work together on a document using streaming video, audio and other meta-data.

Recent improvement in processor speed, network systems, and the availability of massive digital storage has led to an explosive amount of multimedia data online [SO03, HBC$^+$03]. Already, according to network infrastructure company CacheLogic, more than 40 percent of Internet traffic is being taken up by peer-to-peer swaps that involve video content. Add to that the growing amount of legitimate content from companies such as Apple Computer, YouTube, and Google Video, and the scale of consumers' demand for video begins to emerge. However, if all these multimedia data are not manageable and accessible by general users, they will become much less useful in practice. This statement has been reflected in one of the *SIGMM* grand challenges [RJ04]:

> "Making capturing, storing, finding and using digital media an everyday occurrence in our computing environment".

To achieve this, *multimedia information retrieval* systems, which aim to search a large number of multimedia data for documents relevant to an information need, offer an important platform to access and manage the vast amount of multimedia content online. Increasingly, this technique has drawn more and more attention from the extant web search engines (e.g. *Google*, *Yahoo!*, *Blinx.tv* and so on).

Given the luxury of accessing data from multiple streams, the multimedia retrieval community has been gradually shifting its emphasis from analyzing one media source at a time to exploring opportunities to select and combine diverse knowledge sources from correlated media types and context, especially as recent years have seen the rapid development of large-scale semantic concept detection techniques and retrieval approaches on various modalities. For example, searching multimedia collections for “the election of a US president” might need to leverage information from these contexts: restricting the search collections to news video, identifying the segments with persons on the screen and looking for persons speaking the word “election” in the audio stream. In order to develop a knowledge source combination strategy, two basic issues must be addressed, i.e., *what* to combine (i.e., identify available knowledge sources from multimedia data) and *how* to combine (i.e., develop effective combination strategies to merge multiple knowledge sources). Hence in the remainder of this chapter, we will elaborate both issues followed by summarizing the contributions of this thesis.

1.1 Multimedia Retrieval and Multimedia Knowledge Source

Generally speaking, a multimedia retrieval system represents information need as multimodal queries and represents documents as a vector of internal multimedia representations. The queries may merely consist of text or also contain information from multiple modalities such as image, audio or video examples. The document relevance with respect to the query are decided by a set of retrieval status values(RSVs) between each pair of the query and the document representation. Then a ranked list of documents with the highest RSVs is returned to users. Specifically, this thesis considers the task of multimedia retrieval with the following properties,

- The multimodal queries can come from a broad range of domains and without prior knowledge about the query topics. Specifically, this thesis deals with semantic queries, which aim to find high-level semantic content such as specific people, objects, and events, rather than queries attempting to find non-semantic content, e.g. “find the video clips with all blue pixels”. The description of information needs is not limited. For example, users can express their request in text which is then matched against indexed labels from video. If they have an image or video example, it can be used to search for similar visual contents in a "query by example" framework.

- All the approaches and experiments are designed based on *heterogenous multimedia archives* [Wes04], which allow huge variabilities on the topics of multimedia collections. Two examples for heterogenous multimedia archives are news video archives and downloaded video collections from the world wide web. This contrasts with *homogeneous multimedia archives* collected from a narrow domain, e.g., medical image collections, soccer video, recorded video lectures, and frontal face databases.

- In general, users can be categorized into three broad categories [SWS+00]. (1) A general class of users aimed at *browsing* over a large number of documents from diversified sources, where users have no specific target at the beginning except for finding interesting things. This scenario usually implies iterative refinement of the information needs and highly interactive search systems. (2) Another class of users want to do *arbitrary search* by retrieving an arbitrary document satisfying his information need that can be presented by text keywords or visual examples. Arbitrary search usually places more emphasis on precision in top-ranked documents. (3) The third class of users, *complete search/annotation*, aims to discover every relevant document that belong to a specific category. To support these users, the retrieval systems must possess more automatic processing power to reduce the huge manual annotation efforts. In these systems, recall in the entire collection is an important criterion to optimize. This thesis particularly focus on the last two types of users because their information needs are explicitly defined, although our discussions can also be extended to the first type of users once their goal become clearer.

The early multimedia retrieval systems [Lew02, SLN+02, WCGH99] usually model documents with a set of (low-level) detectable features generated from different modalities, such as tf.idf weights for text transcripts and color/texture histograms for images. The top ranked documents are retrieved by finding the most similar documents to the query examples in low-level feature space. However, the effectiveness for these low-level representations is usually limited owing to two reasons. First, text information alone in multimedia documents, e.g., speech transcripts and closed captions, is not predictive enough to support multimedia retrieval. One reason is that relevant video clips might show up when no associated words are spoken in the transcript. A reporter may also speak about a topic with the relevant footage following later in the story, resulting in a major time offset between keywords and the relevant clips. Word sense ambiguity and speech recognition errors may also result in a failure of capturing the relevant documents. More importantly, text information is not always available in multimedia data such as surveillance or soccer video. Second, although retrieval outputs can be augmented with other modalities such as visual features, these low-level representations typically result in the notorious *semantic gap* between users and retrieval systems, due to their inability to capture the semantic meaning of video content. To handle a semantic query, users have to take a detour by either manually translating semantic content into lower-level representations or by providing suitable examples to represent the information needs.

In order to better support semantic-based multimedia retrieval, an intermediate layer of hundreds of *semantic concepts* has been introduced [NKH00] in an effort to capture the semantic content of multimedia documents. The typical concepts includes a wide range of topics such as those related to people(face, anchor, etc), acoustic(speech, music, significant pause), objects(image blobs, buildings, graphics), location(outdoors, city, studio setting), genre(weather, financial, sports) and production(camera motion, blank frames) [CMC05]. The task of automatic semantic concept detection has been investigated by many researchers in recent years [BDF$^+$02, NKFH98, LTS03, YN05b, YCH04, JLM03, WCCS04, SP02, SSL02, VJZ98, VFJZ99]. Their successes have demonstrated that a large number of high-level semantic concepts are able to be inferred from the low-level multi-modal features with a reasonable detection accuracy.

The previous analysis suggests us to construct multimedia retrieval systems using a number of diverse knowledge sources, including low-level features and semantic concepts, as the basic building blocks. Thus, we can define the concepts of knowledge sources, ranking features, and the multimedia retrieval problem as follows:

Definition 1. *(**knowledge source**) The basic building block for combination in the retrieval process, such as a retrieval component built on one type of low-level video feature, or a high-level semantic concept indexed in the database.*

Definition 2. *(**ranking feature**) The retrieval outputs generated from each knowledge source. For example, a ranking feature can be a binary $\{0,1\}$ output from a face concept detector or a real-value [0,1] output from a text retrieval expert. In this way, each multimedia document can be naturally represented as a bag of*

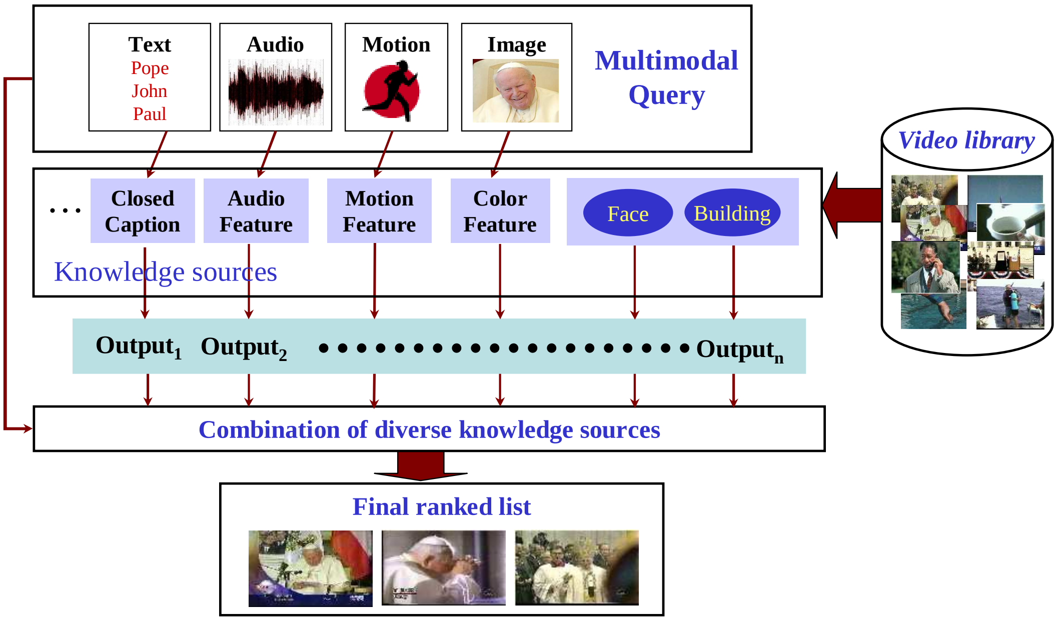


Figure 1.1: Design of multimedia retrieval systems for video data.

ranking features generated from diverse knowledge sources. This can be viewed as a "bag-of-ranking-features" representation.

Definition 3. *(**multimedia retrieval**) Given a multimodal query Q and a collection of multimedia documents $\{D_1, ..., D_{M_D}\}$ where each document is associated with a bag of ranking features generated from different knowledge sources, return a ranked list of documents based on descending order of relevance in the multimedia collection.*

To illustrate, Figure 1.1 shows the design of a typical multimedia retrieval system based on video collections [HBC$^+$03]. First, each video footage is segmented into a number of smaller clips as "documents", and various sets of low-level features are extracted from the video clips through analysis of multimedia sources. Each video clip is then associated with a vector of ranking features, which include both individual outputs from different retrieval experts indicating the query-document similarity from a specific modality, as well as the detection outputs of semantic concepts which can be generated offline. Finally, the system combines these ranking features based on the query description to produce a final ranked list of multimedia documents.

1.2 Knowledge Source Combination

Although a large body of work has been devoted to extracting various ranking features from multimedia sources, relatively less attention has been given to this problem: find a suitable strategy to combine diverse knowledge sources based on user information needs. This task is not only a significant challenge, but also offers great promise to provide considerable improvement in retrieval [YYH04].

Until recently it remained unclear how these heterogenous knowledge sources could be systematically combined in the context of multimedia retrieval. The mainstream approaches rely on query independent combination strategies that can be either predefined as some combination functions or learned from some development sets. However, in these approaches, the combination parameters would not be able to change across various information needs and hence result in a considerable loss of flexibility. For example, the query "finding George Washington" and the query "finding White House" should not share equivalent weights, because the former one prefers the outputs from face recognition and text retrieval whereby the latter one prefers the outputs from image retrieval. Previous experiments [YYH04, CNL$^+$04] demonstrated that with respect to the best query independent weights, the performance of multimedia retrieval could be considerably improved by adapting combination functions according to query variations. This observation motivates us to explicitly incorporate the factor of information need into the design of combination functions.

In this thesis, we propose using a conditional probabilistic retrieval model as the basic framework to combine diverse knowledge sources in multimedia retrieval, which translated the retrieval task into a (supervised) learning problem with the parameters learned discriminatively. In contrast to the typical choices of generative models, a discriminative learning model is suggested for estimating the combination parameters in order to deal with heterogenous ranking features. We also propose an efficient rank learning approach called "ranking logistic regression" that can explicitly model the ranking relations in the learning process with much less computational effort. Based on this probabilistic retrieval model, we developed a two-stage learning approach so that the query information can be modeled in the knowledge source combination with a solid probabilistic foundation. To be more concrete, the proposed approaches contain 1) a query analysis stage which can discover the mixing structure of the query space based on the past relevance judgments and 2) a context analysis stage which can automatically leverage additional ranking features to refine the initial retrieval results.

The query analysis approaches aim to adapt the combination functions for

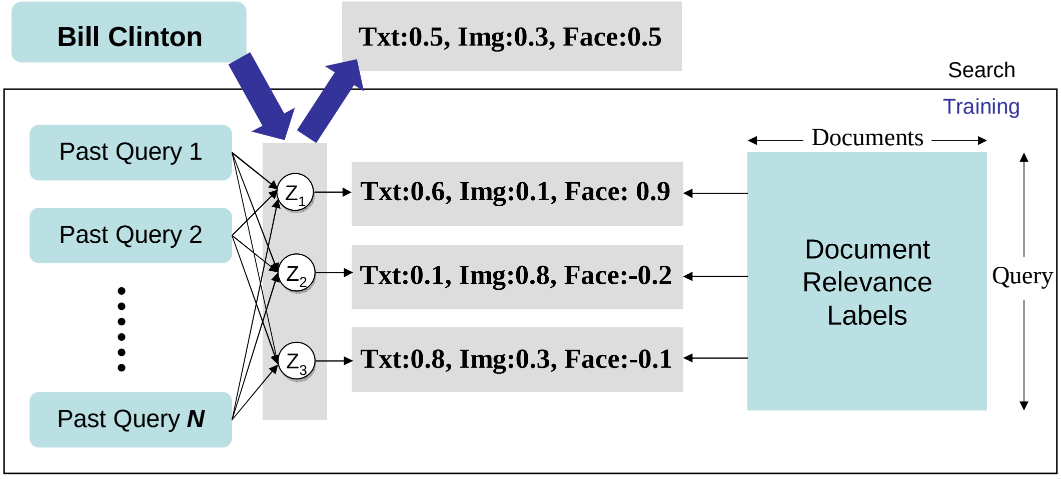

Bill Clinton
Txt:0.5, Img:0.3, Face:0.5
Search
Training
Past Query 1
Past Query 2
Past Query N
Z_1
Z_2
Z_3
Txt:0.6, Img:0.1, Face: 0.9
Txt:0.1, Img:0.8, Face:-0.2
Txt:0.8, Img:0.3, Face:-0.1
Documents
Document Relevance Labels
Query

Figure 1.2: Illustration of query analysis approaches, which can automatically discover the mixing structure of the query space based on previous relevance judgment and predict the combination parameters for unseen queries.

each unseen query by learning from past retrieval results. However, given the virtually infinite number of unseen queries, it is impractical to learn the combination function simply on a per query basis. A feasible alternative is to learn from predefined query classes, i.e., associating combination weights with a few pre-defined classes which consist of queries with similar characteristics.[1] In this case, it is legitimate to collect truth data for each query type because the number of types is very limited, while the learned weights can be reused for other unseen queries as long as they belong to some of the predefined classes. The effectiveness of this method has been demonstrated by previous work [YYH04, CNL$^+$04]. To extend the idea of query-class based retrieval, we also propose an approach called *probabilistic latent query analysis*(pLQA) inspired by the algorithm *probabilistic latent semantic analysis*(pLSI) [Hof99], with the goal of automatically discovering the mixing structure of the query space without explicitly defining query classes. Three pLQA models have been discussed which evolve from a basic version(BpLQA) to an adaptive version (ApLQA) that operates on the query feature space and a kernel version (KpLQA) that builds on a Mercer kernel representation. This formulation offers a probabilistic interpretation for latent query

[1] Its underlying assumptions are: 1) the query space is organized as a finite number of query classes and the queries in each class share the same combination functions and 2) query descriptions can be used to indicate which class the query belongs to.

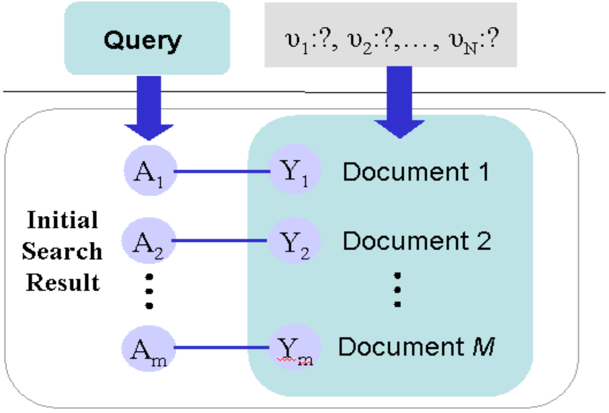


Figure 1.3: Illustration of context analysis approaches, which can automatically leverage additional ranking features to refine the initial retrieval results. It treats the document relevances and the weights of "un-weighted" ranking features as latent variables.

classes, provides guideline to estimate number of query classes, and allows the mixing of multiple query types in a single query. A further extension of pLQA is called *hierarchical pLQA* model (HpLQA), which can model the distributions of query-specific combination components in a single query class via a hierarchical Bayesian model.

However, the effectiveness of query analysis is limited by the amount of training data. As a complementary method, context analysis aims to adapt the combination function specifically to the current query by means of treating the combination weights of ranking features as missing variables. In particular, a *probabilistic local context analysis*(pLCA) model is proposed to automatically leverage useful ranking features to improve the initial retrieval outputs. Formally, it can be described as an undirected graphical model that treats the document relevances and the weights of "un-weighted" features as a set of latent variables. In this model, the marginal dependence between initial retrieval results and latent concept weights allow the usefulness of each semantic concept to be determined in the retrieval process. In the case where human feedback is available, we also propose a pLCA variant to adjust the combination parameters based on human relevance feedback. An approximate inference algorithm is developed to accelerate the parameter updating process.

To evaluate the effectiveness of multimedia retrieval on large scale data collections, the proposed approaches are applied to TRECVID datasets, which contain various queries from broad domains and multiple video collections of heterogenous topics. Well-established evaluation methodologies and metrics are adopted to demonstrate the effectiveness of our methods in the evaluation. Note that the proposed approaches are also applicable in other domains such as combining audio-visual outputs based on the environments, combining multiple search engines based on queries, and combining multiple answers from various sources based on questions. To demonstrate the effectiveness of the proposed approaches outside the multimedia data, we also extend the experiments to a meta-search task that combines the outputs from multiple search engines to form a better ranked list.

1.3 Thesis Overview

The contributions of this thesis can be summarized as follows,

- This thesis provides a principled retrieval framework to combine diverse knowledge sources by using a conditional probabilistic model. This model allows multiple forms of ranking features (query dependent and query independent features) as well as query information and context information to be integrated in a unified framework. Different from previous work, a discriminative learning method is suggested for estimating the combination parameters to deal with heterogenous ranking features. An efficient rank learning approach called "ranking logistic regression" is developed that can explicitly model the ranking relations in the learning process with much less training time.

- This thesis surveys and compares a number of state-of-the-art approaches for extracting ranking features from various multimedia knowledge sources, such as text retrieval, image retrieval and semantic concept detection in large-scale multimedia collections. These studies offer a useful guideline for researchers to select suitable algorithms to deal with different knowledge sources in multimedia systems. Several novel approaches are proposed in this thesis to extract ranking features in a more effective way, e.g., SVM ensembles to handle rare classes, semi-supervised cross feature learning to leverage multimodal information, undirected graphical models to model concept relations, and dual-wing harmoniums to discover hidden concepts. Finally, our case study results confirmed that a few thousand semantic concepts could be sufficient to support high accuracy video retrieval systems.

- To incorporate query information in the combination function, we develop several *query analysis* models which can automatically discover the mixing structure of the query space based on previous retrieval results and predict the combination parameters for unseen queries. In more detail, we propose the *query-class based retrieval* model by manually defining the query classes and a series of *probabilistic latent query analysis*(pLQA) models which can automatically discover latent query classes from the training data.

- To adapt a combination function on a per query basis, we develop a *probabilistic local context analysis model*(pLCA) to automatically leverage useful high-level semantic concepts to improve the initial retrieval outputs. This approach can be represented as an undirected graphical model by treating document relevances and combination weights of semantic concepts as latent variables. We also propose a pLCA variant to adjust the combination parameters based on human relevance feedback.

- All the approaches are evaluated on the multimedia retrieval task with multiple large-scale video collections, which confirms the promising performance of the conditional probabilistic retrieval framework with query analysis and context analysis. We provide a thorough study on how the proposed retrieval approaches perform on standard multimedia collections and offer baseline performances for other researchers to compare with. Our internally developed relevance judgements for other query/document sets will be contributed to the video retrieval community to help develop better algorithms.

- Although the combination approaches developed in this thesis are motivated by the multimedia retrieval problem, their contributions and potential applications are not only limited to this domain. For example, most of the proposed approaches are also evaluated on the task of meta-search over large-scale text collections. The applicability of the proposed methods can be extended to many other areas such as question answering, web IR, cross-lingual IR, multi-sensor fusion, human tracking and so forth.

The rest of the thesis is organized as follows. Chapter 2 reviews the related combination approaches in the fields of multimedia retrieval, ad-hoc text retrieval, web retrieval, and data mining. It also provides background material for the Informedia project and the annual TRECVID video retrieval evaluation. Chapter 3 surveys the general approaches for extracting ranking features, and compares their performance on video collections. Chapter 4 presents the relevance-based probabilistic retrieval model as our basic retrieval framework and a discriminative learning method to estimate the combination parameters. Chapter 5 and Chapter 6 describe the query analysis approaches and context analysis approaches respectively, with both of them supported by experimental evaluations on large-scale video collections. Chapter 7 presents the experimental results of meta-search on large-scale text collections. Finally, Chapter 8 concludes the thesis and envisions the future directions of multimedia information retrieval.

Chapter 2

Literature Review

Knowledge source combination has proved to be a useful and powerful paradigm in several computer science applications including multimedia retrieval [YYH04, WIB$^+$03, KNC05], text information retrieval [SF94, AM01], web search [DKNS01, RS03], combining experts [CSS98], classification [FKS03], database [FLN01] and biological collection [SM01]. In this section, we first review some related approaches in the context of multimedia retrieval, followed by reviewing more related work from other research areas such as meta-search and web retrieval. Finally, we provide the background materials for the Informedia project and the NIST TREC video track (TRECVID) [SO03] by describing its tasks, evaluation methods and collection statistics in detail.

2.1 Combination in Multimedia Retrieval

Designing the combination approaches for multiple knowledge sources is of great importance to develop effective multimedia retrieval systems. As pointed out by Manmatha [Man02], this task remains an important challenge for researchers,

> “ To deal effectively with multimedia retrieval, one must be able to handle multiple query and document modalities. In video, for example, moving images, speech, music audio and text(closed captions) can all contribute to effective retrieval. Integrating the different modalities in principled ways is a challenge. ”

The problem of multimedia knowledge source combination has been actively investigated in recent years. Westerveld et al. [WIB$^+$03] demonstrated how the combination of different models/modalities can affect the performance of video retrieval. They adopt a generative model inspired by language modeling approach and a probabilistic approach for image retrieval to rank the video shots. Final results are obtained by sorting the joint probabilities of both modalities. The video retrieval system proposed by Amir et al. [AHI$^+$03] applied a query-dependent combination model that the weights are decided based on user experience and a query-independent linear combination model to merge the text/image retrieval systems, where the per-modality weights are chosen to maximize the mean average precision score on development data. Gaughan et al. [GSG$^+$03] ranked the video clips based on the summation of feature scores and automatic speech retrieval scores, where the influence of speech retrieval is at four times that of any other features. Rautiainen et al. [Rea04] use a Borda-count variant to combine the results from text search and visual search. The combination weights are pre-defined by users when the query is submitted. The QBIC system [FBF$^+$94] combines scores from different image techniques using linear combination. Fagin [Fag98] used standard logical operators like MIN and MAX to combine scores in a multimedia database.

However, until recently most of the multimedia retrieval systems use query-independent approaches to combine multiple knowledge sources. This has greatly limited their flexibilities and performance in the retrieval process [YH03]. Instead, it is more desirable to design a better combination method which can take query information into account without asking for explicit user inputs. Recently, query-class based combination approaches [YYH04, CNL$^+$04] have been proposed as a viable alternative for the query independent combination, which begins with classifying the queries into predefined query classes and then applies the corresponding combination weights for knowledge source combination. Experimental evaluations have demonstrated the effectiveness of this idea, which have been applied in the best-performing systems of TRECVID manual retrieval task [SO03] from the year of 2003. Also, the validity of using query-class dependent weights has been confirmed by many follow-on studies [CNG$^+$05, Huu05, YXW$^+$05, KNC05]. For example, Huurnink [Huu05] suggested it is helpful to categorize the queries into general/special queries and simple/complex queries for combination. Yuan et al. [YXW$^+$05] classified the query space into person and non-person queries in their multimedia retrieval system. To improve upon the manually defined query classes, Kennedy et al. [KNC05] recently proposed a data-driven learning approach to automatically discover the query-class-dependent weights from training data by means of grouping the queries in a joint performance and semantic space via statistical clustering techniques such as hierarchical clustering and k-means.

A more recent work [YH06c] unified query class categorization and combination weight optimization in a single probabilistic framework by treating query classes as latent variables.

An alternative approach for multimedia retrieval is to use text-based structural data retrieval techniques to search a "big" structural data representation that includes all the information of textual features and semantic concepts. The "Multimedia Content Description Interface"(MPEG-7) [Smi03] is the most widely adopted storage format for video retrieval.[1] A number of successful video retrieval systems have been built upon the MPEG-7 representation. For instance, a MPEG7 framework to manage the data in audio-visual representation is proposed in [TFC03]. The annotation is based on a fixed domain ontology from TV-Anytime and the retrieval is restricted to querying the metadata for video segments. Graves et al. [GL02] proposed an inference network approach for video retrieval. The document network is constructed using video metadata encoded using MPEG-7 and captures information about different aspects from the video. To provide more semantic and reasoning support for the MPEG7 formalism, a framework for querying multimedia data using a tree-embedding approximation algorithm as well as reasoning through an ontology has been proposed [HBHV04]. Generally speaking, the knowledge sources provided by textual features, imagery features and semantic concepts are supposed to be treated differently in these text-based approaches. But so far, most previous work simply uses query-independent combination functions or requires users to adjust the weighting. We believe automatically learning a good combination strategy over diverse knowledge sources would be a valuable complementary tool to augment the MPEG7-based video retrieval systems.

Multimodal combination is also an active research area for extracting high-level semantic concepts from multimedia streams. In this task, it is reasonable to either combine multiple detection models, combine the detection models with different underlying features, or combine the models with the same underlying features but different parameter configurations. Among them, the simplest methods are those fixed combination approaches. For instance, Amir et al. [AHI$^+$03] studied several multi-modality and multi-model fusion method for the TRECVID semantic feature extraction task based on min, max and linear combination. It demonstrated the importance of fusion for achieving good performance for video

[1] MPEG7, formally named "Multimedia Content Description Interface", is an ISO/IEC standard that aims at describing the multimedia data content by attaching them to the metadata based on XML schema.

annotation. However, in order to gain further improvement, a large body of machine learning methods have also been proposed and studied. For instance, a more advanced multi-modal fusion strategy [WCCS04] called super-kernel fusion has been proposed, where the underlying idea is to construct a hierarchy of kernel machines to enable of the modeling the complex decision boundaries beyond the linear combination. Yang et al. [YCH04] specifically considered the problem of detecting news subjects in news video archives by linearly combining the multimodal information in videos, including transcripts, video structure and visual features. Cees et al. [SWS05] compared the early fusion and late fusion methods by using SVMs as the base classifiers and meta-level classifiers for text-image fusion. Their experiments on 184 hours broadcast video and 20 semantic concepts shows that late fusion tends to get slightly better performance than early fusion for most concepts, but when the early fusion scores better than one concept, the improvement will be more significant.

2.2 Combination in Text Retrieval

Text information retrieval is the task of searching a static text collection for relevant documents given a short-term information need. Typical examples of adhoc text retrieval are the common web search engines, e.g. *Google*, *Altavista*. For text retrieval, queries and documents are generally described using independent word features. A common example for query/document representation is the tf*idf weights for bags of words.

Multimedia retrieval is closely related to the text retrieval tasks that need to combine outputs from different data sources. Examples include meta-search [SF94, MFR01] that merges results from multiple databases and web retrieval that fuses hyperlink scores with content-based retrieval scores [Kle98]. The query-dependent features such as uni-modal retrieval outputs in the task of multimedia retrieval can be analogous to the single source search outputs in the meta-search. Similarly, the query-independent features such as the high-level semantic concepts can be analogous to the hyperlink-based features (e.g. PageRank scores) in the web retrieval. Table 2.1 parallels the corresponding concepts between multimedia retrieval and ad-hoc text retrieval. As shown in the table, concepts in both domains mostly correspond to each other, which indicates that their retrieval methods could bear lots of similarities. However, because of the fundamentally different representations in the multimedia documents, the retrieval algorithms in text retrieval are not always applicable in the multimedia retrieval task. Therefore,

	Ad-hoc Text Retrieval	**Multimedia Retrieval**
Retrieval Unit	Documents	Multimedia documents
Query Representation	Keywords	Multimodal queries
Low-level Features	Word features	Text/visual/audio features
Mid-level Features	Single source retrieval outputs Query-independent features	Uni-modal retrieval outputs Semantic concepts

Table 2.1: Comparing the terminology between multimedia retrieval and ad-hoc text retrieval

in the following discussions, we present some related work on meta-search and web-based retrieval, and the discuss their connections/differences to multimedia retrieval.

Meta search is a task of retrieving and combining the information from multiple sources, which is usually investigated in two different forms [ACM$^+$02],

- Data fusion: The combination of multiple search engine runs over an effectively common data set for a given query. A typical data set is the World Wide Web(WWW).
- Collection fusion or distributed retrieval: the combination of information from multiple sources that index on a disjoint data set. The typical data sets are specialized data sets such as the U.S. Government Printing Office(GPO) portal.

Data fusion problems usually assume that multiple search engines can provide relevance scores over a common data set. Two main issues of data fusion are described as follows: *relevance score normalization* that converts the outputs of various search engines to a comparable score space, and *score combination* that merges the normalized scores into a final output. To normalize the relevance scores across difference search engines, the common approaches are either a linear mapping of the scores into a range of [0,1], or a distribution-based normalization [MFR01] by fitting the distributions of relevant/irrelevant documents on a per query basis. The normalized scores could be combined via different methods. Shaw et al. [SF94] proposed a number of combination techniques named COMB{MIN, MAX, ANZ, MED, SUM, MNZ}. The best performing strategies are among COMBSUM (equivalent to averaging), i.e., taking the sum of scores, and COMBMNZ (equivalent to weighted averaging) , i.e., multiplying this sum by the number of engines that have non-zero scores. Vogt and Cottrell [VC99] experimented with the weighted linearly combination to merge the relevance scores from underlying search engines. The learned weights are independent of the queries. Aslam et al. [AM01] proposed a probabilistic model based on Bayes inference using rank information instead of scores, which achieves good results but requires extensive training efforts. They also studied a rank-aggregation approach called the (weighted) Borda-fuse stemmed from Social Choice Theory. Ogilvie [OC03] studied the approaches of several data fusion algorithms and mixture-based language models for the task of known-item search. However as pointed by Aslam et al. [ACM$^+$02], one of the major challenges for the data fusion problem so far is that the variation of information needs has seldom been modeled into the combination functions,

> “ The performance of search engines vary from query to query. ... Can a meta-search technique be developed which consistently outperforms the best underlying search engine? Or can a technique be developed which is capable of distinguishing the “good” underlying system from “bad” on a per query basis? ”

Result merging in distributed information retrieval, i.e. combining the retrieval information from multiple resources, is different from the data fusion problem in the sense that the contents exploited from different information sources are usually independent which results in much less overlap information among individual ranked lists [SC02]. In an uncooperative environment, we cannot assume the information sources use the same type of retrieval algorithms and can provide accurate frequency information. Voorhees et al. [VGJL95] used query clustering techniques to detect similar queries in a training query set, applied learned parameters to select top documents from each sources and combined the final results by a (weighted) round robin method. CORI [CCB95] associated a normalized value to each information source. It showed that the weighted linear combination with source values can achieve considerably better results than interleaving ranks. A logistic transformation model [CS00] was proposed to build logistic models for all the information sources using human-judged training data and transform information-source specific scores into source independent scores. Lin et al. [LH04] have applied a similar strategy for fusing various outputs of news video collections and achieved considerable improvement over the round-robin combination. Note that the above methods are all independent of the query descriptions. Si et al. [SC02] considered a query-specific combination using a semi-supervised learning approach. It linearly merges the retrieval results based on a centralized sample database, which outperformed the CORI method in various datasets.

Some of the meta-search techniques can be naturally applied in the context of multimedia retrieval, such as COMB$\{$SUM, MNZ$\}$, Borda fuse and weighted linear combination. However, the scenarios of multimedia retrieval bear some major differences with meta-search and impose different requirements on the combination strategies. First, the semantic concepts are query-independent features, which do not provide a ranked list of documents in the descending order of relevance. This property makes it less desired to discard the query information in the combination function. Moreover, we do not have a central database as a criterion to adjust the retrieval scores across different sources.

2.3 Combination in Other Research Areas

Web retrieval is another related research domain. The common web search engines fuse the evidence from content information and hyper-link information in an attempt to compensate the weakness of either type of knowledge. In the hubs and authorities model proposed by Kleinberg [Kle98], the content-based scores are implicitly combined with the link-based scores, which are computed locally within the top-ranked documents. PageRank [BP98] is another type of link-based scores which can be pre-computed from the link structure of the web documents using “random surfing” technique. The authors [BP98] combine PageRank with the content-based scores using some monotonic functions while the details are not provided. Kang et al. [KK03] discovered the fact that the best combination strategies vary with the query types and proposed to use query types by selecting combination functions for web document retrieval.

In the community of data mining and database applications, the problem of knowledge source combination is widely investigated under the name of *rank aggregation*, which aims to combine multiple ranking results from various sources. These studies usually considered using query-independent combination schemes, while they are more interested in improving the efficiency to find top few examples from multiple partial ranking lists. For instance, the threshold algorithm [FLN01] is an elegant and simple aggregation algorithm which can achieve optimality over every database in terms of the middleware cost incurred by the algorithm. This algorithm requires only a small and constant-size buffer to implement. Dwork et al. [DKNS01] proposed a rank aggregation and an efficient approximation approach by finding a ranking whose total Kendall tau distance to the given rankings is minimized, where the Kendall tau distance is defined as the number of pairwise disagreements between two rankings. Cohen et al. [CSS98] considered the problem of learning how to order the instances based on the feedback of relative preference judgments. Since the exact inference algorithm is NP-complete, they presented an greedy algorithm which is guaranteed to find a good approximation. They also proposed an online learning algorithms to find a combination function of ranking experts.

2.4 Background I: The Informedia Digital Video Library System

The Informedia Digital Video Library [HW96] project focuses specifically on information extraction from video and audio content. Over two terabytes of online data have been collected in MPEG-1 format, with metadata automatically generated and indexed for retrieving videos from this library. The architecture for the project is based on the premise that real-time constraints on library and associated metadata creation could be relaxed in order to realize increased automation and deeper parsing and indexing for identifying the library contents and breaking it into segments. Library creation is an offline activity, with library exploration by users occurring online and making use of the generated metadata and segmentation.

The Informedia research challenge is how the information contained in video and audio can be analyzed automatically and then made useful for a user. Broadly speaking, the Informedia project wants to enable search and discovery in the video medium, similar to what is currently widely available for text. One prerequisite for achieving this goal is the automated information extraction and metadata creation from digitized video. Once the metadata has been extracted, the system enables full-content search and retrieval from spoken language and visual documents. The approach that has been most successful to date, involves the integration of speech, image and natural language understanding for library creation and retrieval.

The Informedia interface provides multiple levels of summaries and abstractions for users: visual icons with relevance measure [CM98], short titles or headlines [JH01], topic identification of stories [HL98], filmstrip (storyboard) views [CHWC99], transcript following [HW96], dynamic maps [COH00], active video skims [CHWC99], face detection and recognition [Hou99], image retrieval [FBF$^+$94, HP02]. Once a relevant video clip has been found, a user might want to make annotations for herself or others to later reuse what was learned. The Informedia system provides a simple mechanism that allows a user to type or speak any comment that applies to a user-selected portion of the video. To this end, the indexing mechanism was modified to allow dynamic, incremental additions and deletions to the index. Finally, a fielded search capability enables the user to search only on selected fields, for example searching only the user annotation field, either through a statistical search based on the OKAPI BM-25 formula or with a classic Boolean search expression. Further re-use of relevant video clips is enabled through a cut-and-paste mechanism that allows a selected clip to be extracted from the library and imported into PowerPoint slide presentations or MS Word text documents.

2.5 Background II: TREC Video Track

The National Institute of Standards and Technology(NIST) has sponsored the annual *Text REtrieval Conference* (TREC) as a mean to encourage research within the information retrieval community by providing the infrastructure and benchmark necessary for large-scale evaluation of retrieval methodologies. In 2001, NIST started the TREC Video Track (now referred to as TRECVID [SO03]) to promote progress in content-based video retrieval via an open, metrics-based evaluation, where the video corpora have ranged from documentaries, advertising films, technical/educational material to multi-lingual broadcast news. The international participation of TRECVID has rapidly grown from 12 to 62 organizations and academic institutions from 2001 to 2005.

The TRECVID forum has defined a number of tasks including shot detection, story segmentation, semantic feature extraction and automatic/manual/interactive search. Among them, the search tasks in TRECVID are the extensions of their text analogues from previous TREC evaluations on pure text documents. Participating groups are required to index a given test collection of video data and return lists of relevant clips from the test collection. The search topics are designed as multimedia descriptions of an information need, which might contain not only text keywords but also possibly video, audio and image examples. Typically, the topics include requests for some specific items, specific people, specific facts, instances of categories and instances of activities. In analogy to “document” in text retrieval, TRECVID adopts the basic video units to be retrieved as video shots, which is defined as a single continuous camera operation without an editor’s cut, fade or dissolve. To simplify ranking, the rank lists can only contain up to N shots relevant to the query where N=100 for 2002, N=1000 for 2003 and 2004.

1. Find shots of Yasser Arafat.
2. Find shots of a rocket or missile taking off.
3. Find shots of the Tomb of the Unknown Soldier at Arlington National Cemetery.
4. Find shots of the front of the White House in the daytime with the fountain running.

Figure 2.1: Text query examples from TRECVID 2003.

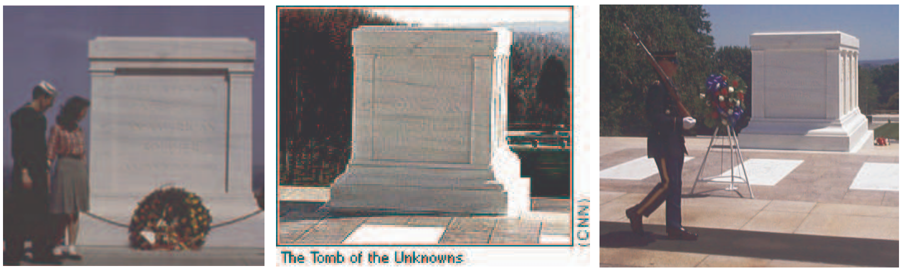


Figure 2.2: Image query examples of “Find the Tomb of Unknown Soldiers at Arlington National Cemetery”.

The shot boundaries and the ground truth of search results are officially provided. The ground truth is pooled from all participants’ submission.

The search task distinguishes between interactive approaches, where a user can interact with the system repeatedly to locate the relevant shots, manual approaches, in which a user is only allowed to modify the query before submitting it to the retrieval system, and finally fully automatic approaches, where systems must directly parse the queries and provide relevant results. *Precision* and *recall* are two central criteria to evaluate the performance of retrieval algorithm. Precision is the fraction of the retrieved documents that is relevant. Recall is the fraction of relevant documents that is retrieved. NIST also defines another measure of retrieval effectiveness called non-interpolated average precision over a set of retrieved documents (shots in our case). Let R be the number of true relevant documents in a set of size S; L the ranked list of documents returned. At any given index j let R_j be the number of relevant documents in the top j documents. Let $I_j = 1$ if the j^{th} document is relevant and 0 otherwise. Assuming $R < S$, the non-interpolated average precision **(AP)** is then defined as $\frac{1}{R}\sum_{j=1}^{S}\frac{R_j}{j} * I_j$. Mean average precision(MAP) is the mean of average precisions over all queries.

collection	# queries	# shots	duration	explanation
t02s	25	24,263	40.12h	TREC'02 search set
t03d	25	47,531	62.20h	TREC'03 development set
t03s	25	75,850	64.30h	TREC'03 search set
t04d	24	124,097	127.00h	TREC'04 development set
t04s	24	48,818	70.00h	TREC'04 search set
t05d	24	74,532	80.00h	TREC'05 development set
t05s	24	77,979	80.00h	TREC'05 search set

Table 2.2: Labels of TRECVID video collections and their statistics.

As one of the largest publicly available video collections with manual annotations, TRECVID collections have become the most used standard large-scale testbeds for the task of video retrieval. Each video collection of TRECVID'03 - '05 is split into a development set and a search set chronologically by source. The development sets are collaboratively annotated by the participants and used as the training pool to develop automatic multimedia indexing/retrieval algorithms in low-level video feature extraction, high level semantic feature extraction and search tasks. The search sets mainly serve as testbeds for evaluating the performances of retrieval systems. In Table 2.2, we provide the statistics for all of the data collections and their abbreviations used in the following experiments. We also provide additional information of these TRECVID video collections as follows where the details of query topics are listed in the Appendix,

- **TRECVID2002**: In this collection NIST defined 25 search topics to find within a search test collection of 40.12 hours of video from the Prelinger Archives and the Open Video archives. The material consists of advertising, educational, industrial, and amateur films produced between the 1910s and 1970s. The search test collection was delineated into 14,524 video shots as the common shot reference.

- **TRECVID2003**: In this collection NIST defined another 25 search topics and provide 120 hours (240 30-minutes programs) of ABC World News Tonight and CNN Headline News recorded by the Linguistic Data Consortium from late January through June 1998, along with 13 hours of C-SPAN programming from mostly 2001. Among them, 6 hours of video are used for shot boundary detection, and the reminder are split into a development set including 47,531 shots representing 62.2 hours of video and a search set including 75,850 shots representing 64.3 hours of video.

- **TRECVID2004**: In this collection 25 search topics are defined. NIST provided a new set of approximately 70 hours (48,818 shots) of video as the search set, captured by the Linguistic Data Consortium during the last half of 1998 from both CNN Headline News and ABC World News Tonight. The data is used with the TRECVID2003 data (as the development set) so that investments in annotating and truthing the news genre can be reused and iteratively improved. The Informedia project [WCGH99] provided a large set of low-level features for the 2004 development data as a common reference for TRECVID researchers

- **TRECVID2005**: In this collection 24 search topics are defined. The video collection includes a 170-hour (nearly 150k shots) multilingual news video captured from MSNBC (English), NBC Nightly News (English), CNN (English), LBC(Arabic), CCTV(Chinese) and NTDTV (Chinese). Among them, 6 hours of video are used for shot boundary detection, and the reminder are split into a development set including 74,532 shots representing 80 hours of video and a search set including 77,979 shots representing another 80 hours of video.

Chapter 3

Generation of Multimedia Ranking Features: A Survey

In this chapter, we describe and compare a wide range of ranking feature extraction approaches for multimedia data. Specifically, ranking features can be categorized into two types: query-dependent and query-independent features. The query dependent features are usually generated amidst the retrieval process by uni-modal retrieval components on low-level video features. They are used to indicate the similarity between query keywords/examples and multimedia documents in terms of a specific modality. Two examples of query dependent features are text retrieval outputs over the closed captions and image retrieval outputs over the color histogram. In contrast, the query independent features can be extracted and indexed in the databases before the retrieval process. The most widely used query independent features for multimedia retrieval are the indices of a manually defined list of high level semantic concepts. The pool of semantic concepts usually covers a wide range of topics including objects, sites, events, specific personalities, named entities and so on.

Therefore, this chapter mainly describes and discusses the components of text retrieval, image retrieval and semantic concept extraction in the state-of-the-art (automatic) multimedia retrieval systems. Experiments are carried out in the TRECVID data collections in order to provide a fair comparison of different algorithms. We also discuss several open research directions for the task of semantic concept detection, such as the issues of balancing training examples, leveraging unlabeled data, modeling relationship between concepts and extracting hidden concepts without manual labeling. Note that our discussion does not aim to completely cover the entire area of multimedia retrieval, neither does it intend to present an exhaustive survey on all of its individual components. Instead, our goal is to present a broad overview of the major building blocks underlying the multimedia retrieval systems, and provide an series of experimental results to justify the advantages and disadvantages of various feature extraction approaches.

3.1 Text Retrieval

As one of the most important retrieval components for video retrieval, the text retrieval module aims to retrieve a number of top-ranked documents based on the similarity between query keywords and documents' textual features. The textual features can be extracted from a number of information sources such as speech transcript, closed captions and video optical character recognition(VOCR). Unless textual features are not available in the video data (such as surveillance video), they are usually the most reliable sources to handle semantic queries in video retrieval systems [HC04], because the text representation is arguably the best vehicle to convey the semantic information among all of the modalities. In this section, we describe several key components of the text retrieval module, including retrieval models, text sources, temporal expansion strategies and query expansion approaches. For each component, we discuss the strengthes/weaknesses and evaluate the retrieval performance of its possible configurations using the TRECVID data collections.

3.1.1 Retrieval Models

The state-of-the-art text retrieval algorithms typically fall into one of the following two categories, i.e., vector space models and probabilistic models. In the vector space models, the relevance between a document d and a query q is defined based on a distance measure on a very high-dimensional term vector space. To achieve this, it is necessary to convert text documents and queries into a space where each dimension corresponds to an indexed term in the vocabulary. The query-document similarity can then be computed based on predefined distance metrics. In contrast, the classical relevance-based probabilistic models consider the differences of term distribution between the classes of relevant documents and non-relevant documents. For each query term, the term weights are calculated according to the probability that this query term is present in the relevant documents. Beyond the relevance-based probabilistic model, language-model based probabilistic retrieval models have burgeoned over the last decade. They are based on the assumption that a query can be generated by the combination of a language model estimated on the current document and a smoothed model estimated on the entire document collection. Previous experiments [PC98] have shown that vector space models and probabilistic retrieval models perform roughly on par with each other. Instead of being exhaustive in investigating all choices of retrieval models, we select some of the most representative retrieval models with typical parameters to evaluate their performances in the video collections. A more complete survey of text retrieval models can be found in [Kra04].

Vector Space Model

Formally, the vector space model represents each document D_k (the k^{th} document) and query Q as a vector w.r.t. indexed terms,

$$D_k = [d_{k0}, d_{k1}, ..., d_{kW}], Q = [q_0, q_1, ..., q_W],$$

where d_{ki} is the weight associated with the i^{th} indexed term for D_k and q_i is the weight associated with the i^{th} term for Q. Typically the term weights are very sparse and thus most of these vector elements are equal to zero. Based on these term-weight representations, the query-document similarity in the vector space models can be computed using the inner product between their term weights,

$$sim(Q, D_k) = \sum_{i=1}^{W} q_i d_{ki}.$$

The simplest approach to design the term weights is to associate each term with a binary 0/1 weight according to the word presence and absence status. However, term weights do not have to be restricted to the binary representation. Instead, in principle the vector of term weights can take any positive real values that encode occurrence and distribution information of the indexed words. For example, one of the most popular term weighting schemes is called $tf.idf$ [Sal89], which suggests the term weights are proportional to the frequency of the term occurrence within the document and inversely proportional to the number of documents where the terms appear. The first factor tf, referred to as term frequency, captures the intuition that if a term is mentioned many times in the documents, it is likely to be a content word and thus important in judging the document relevances. But simple tf will also over-emphasize the impact of some frequently occurring non-content words (a.k.a. stopwords), e.g. determiners, prepositions and auxiliaries. Therefore, the second factor idf, referred to as inverse document frequency, is introduced based on the observation that the more a term appears in different documents, the less discriminative power it has for the retrieval system. Usually, the idf term is converted to its logarithmic value to make it comparable to the tf term. By multiplying these two factors together, we can derive the following similarity measure,

$$sim(Q, D_k) = \sum_{i=1}^{W} q_i^{tf} \cdot q_i^{idf} \cdot d_{ki}^{tf} \cdot d_{ki}^{idf}.$$

Note that the weighting schemes for queries and documents can be different. In fact, selecting specific weighting for queries and documents individually can achieve better retrieval results than a uniform weighting scheme because of the distinctive properties between query descriptions and document description on several aspects, such as document length, number of stopwords and term frequency.

Another dimension of designing term weights is the document length normalization scheme that attempts to eliminate the effects of heterogenous length distribution of text documents. They are useful tools for text retrieval, because if no document length normalization are applied, the retrieval results tend to be biased to long documents that contain more content words to match than short documents. Two of the most well-known document length normalization schemes are the cosine similarity normalization [BW99] and the pivoted document length normalization [SBM96]. The cosine similarity normalization divides the term weights of k^{th} document by a factor of $\sqrt{\sum_i d_{ki}^2}$ so that the sum of squared term weights are normalized to 1. It is called $cosine$ similarity because this normalization transforms the inner product between queries and documents to be the cosine of the angle between query vectors and documents vectors. Despite the intuitive explanation, it was found that the cosine normalization scheme was not always optimal in practical datasets like TREC collections [BW99]. One problem is that long documents in these collections contain many unique terms with misspellings and thus they become unretrievable due to the low term weights. Cosine normalization has another problem in the case where the collection are kept dynamically updating, because the normalization factor needs to be modified every time when the new documents are added to the collection. In fact, the original idea of cosine similarity has been discarded in the current state-of-the-art vector space models [Kra04]. As a better alternative, pivoted normalization proposed by Singhal [SBM96] intends to normalize the document vectors by a factor of $(1 - s)p + sV_d$, where V_d is the document length and two other parameters are slope s and pivot p. It is designed in a way to boost the retrieval scores of short

	code	term weights	meaning
Term Frequency	b	1	Binary word presence
	n	tf	Raw term frequency
	a	$0.5 + 0.5\frac{tf}{\max(tf)}$	Normalized term frequency
	l	$1 + \log(tf)$	Logarithm of raw tf
Inverse Document Frequency	n	1	No idf weighting
	t	$\log(N/n)$	Raw idf weighting
	p	$\log \frac{N-n}{n}$	Probabilistic version of idf weighting
Normalized Factor	n	1	No normalization
	c	$\sqrt{\sum_i d_{ki}^2}$	cosine normalization
	u	$(1-s)p + sV_d$	Pivoted normalization

Table 3.1: The SMART codes for vector space models

documents and decrease the scores of long documents. The parameters of pivoted normalization can be pre-defined or learned on a previous collection. This type of normalization has been proved to be successful in practice although the motivation is not as elegant as cosine normalization.

In order to represent the variants of term weight schemes concisely, previous work always described various term weight schemes with a six-letter code[BW99]. Each of these letters refers to the method of term frequency weighting, the inverse document frequency weighting and the document length normalization of the documents and the query. Table 3.1 lists the letter codes and their explanations in the SMART system. In the following experiments, we also use the same six-letter codes to describe the vector space models.

Probabilistic Models and Okapi Models

In contrast to the vector space models which have a less elegant theoretical basis, the probabilistic models provide a more principled framework by translating information retrieval into an uncertainty inference problem. The underlying principle using probabilistic models for information retrieval is called the *Principal Ranking Principle* [Rob77], which suggests sorting the document D by the log-odds of the document relevance. If we define y as the binary relevance variable where $y = 1$ means the document D is relevant to Q and vice versa, the log-odds ratio can be defined as $\log \frac{P(y=1|D,Q)}{P(y=-1|D,Q)}$. Given this general principle, we can categorize the probabilistic models that have been studied before into three classes,

- *Probabilistic relevance model* [RJ77, RWHB^{+}92]: document relevance is directly estimated given different distributions of indexed terms in relevant documents and irrelevant documents;

- *Inference based model* [Tur91]: the retrieval problem is formulated as a Bayesian inference network;
- *Language-model based model* [PC98, ZL01]: they model the probability that the query can be generated by a statistical language model on the given document.

The simplest probabilistic relevance model is the binary independence retrieval(BIR) model [RJ77]. In this model, each document is represented as a binary vector of the term presence/absence where all the information of term frequencies is discarded. To estimate the term weights, BIR proceeds by inverting the position of y and D based on the Bayes rule and estimating the generative probabilities of document D in the relevant and irrelevant documents,

$$O(y|D,Q) = O(y|Q)\frac{P(D|y_+,Q)}{P(D|y_-,Q)} \propto \prod_{i=1}^{N_q} \left(\frac{p_i}{r_i}\right)^{t_i} \left(\frac{1-p_i}{1-r_i}\right)^{1-t_i} \qquad (3.1)$$

where N_q is the number of the query terms, t_i is the binary indexing for the i^{th} term t_i, p_i is the generative probability given relevant documents $P(t_i = 1|y_+)$ and r_i is the generative probability given irrelevant documents $P(t_i = 1|y_-)$. The last step of Eqn(3.1) can be derived under the independence assumption between retrieved documents and indexed terms. The basic BIR model can only be applied after the parameters p_i and r_i are already estimated for all the query terms.

Clearly, the BIR model has its own limitations, e.g., it only considers the binary presence of indexed terms with frequency information discarded. To address this, Robertson and Walker[RW94] proposed a new probabilistic model based on the 2-Poisson distributions. In this way, it can handle more complex distributions of the query terms attributed to flexibilities of the Poisson distributions. Unfortunately, early experiments that attempted to directly estimate the parameters from training data yields unsatisfying retrieval results, mainly because there are insufficient training data available to estimate such a large number of parameters. Therefore Robertson and Walker considered a series of approximations to the 2-poisson model and finally proposed a series of retrieval models named the Okapi models. Among them, the best known Okapi model is the BM25 version [RWHB$^+$92],

$$\sum_{i=1}^{N_q} \left(s_1 s_3 \cdot \frac{d_{ki}^{tf}}{d_{ki}^{tf} + K^c} \cdot \log \frac{N-n+0.5}{n+0.5} \cdot \frac{q_i^{tf}}{q_i^{tf} + k_3} \right) + k_2 \cdot |Q| \frac{\Delta - d}{\Delta + d}, \qquad (3.2)$$

where $K^c = k_1((1-b) + bd/\Delta)$, $k_1, k_2, k_3, s_1, s_3, b$ are the predefined constants that have to be decided empirically, d is the document length, Δ is the average document length, $|Q|$ is the number of query terms, n is the number of documents containing the query term q_i and N is the size of document collections. It can be found that several tuning parameters in Eqn(3.2) are needed to be determined empirically. Researchers have come up with a simplified retrieval model as follows by setting $k_2 = 0$, $b = 0.75$, $k_1 = 2$, $s_1 = 1$, $s_3 = k_3 + 1$, $k_3 = \infty$ and leaving out the document length correction component,

$$\sum_{i=1}^{N_q} \left(q_i^{tf} \cdot \frac{d_{ki}^{tf}}{d_{ki}^{tf} + 2 \times (0.25 + 0.75 \times \frac{d}{\Delta})} \cdot \log \frac{N - n + 0.5}{n + 0.5} \right), \tag{3.3}$$

This variant, also known as the SMART version of BM25 model[Kra04], has been widely applied in previous work. The document length normalization in this variant reflect the intuition that the longer the text document, the greater the likelihood that a particular query term will occur by chance. More details about the Okapi retrieval models can be found at [RWHB$^+$92].

Recently, statistical language modeling approaches [PC98, ZL01] have emerged as a new probabilistic model for information retrieval, which stems from its earlier counterparts in the field of speech recognition. In this approach, each document is associated with a language model which is a probability distribution over terms. The document ranking is determined by the conditional probability of the query given the language model of documents. Formally, the conditional probability of query Q given a document D is defined as,

$$P(Q|D) = P(q_1, q_2, ..., q_n|D) = \prod_{t=1}^{N} P(q_j|D),$$

where the last step is derived based on the assumption of query term independence given the document language model. Unlike the probabilistic relevance models, the language modeling approach usually estimates the probability of queries conditioned on documents instead of in a reverse way. One justification is that a language model estimated on documents with more term appearances can be more stable than that estimated on the query. Through this model reformulation, language modeling approaches are able to circumvent the problem to explicitly estimate the model of relevant documents in the probabilistic relevance model. Typically, language models from documents could be built efficiently and its performance is demonstrated to be on par with the vector space model. Although sometimes lack of the notion of relevance in the language modeling approaches is a setback for some applications such as relevance feedback, Lafferty et al. [LZ03] showed that the relevance concept can be represented as an implicit variable in modeling and thus what we are actually estimating is $P(Q|D,R)$. To put another way, language modeling and relevance-based probabilistic models are actually two sides of the same coin. Since language modeling retrieval approaches provide such a formal framework for information retrieval, they have been successfully applied in several other information retrieval tasks than text retrieval. For example, these approaches have gained its success in multimedia retrieval [Wes04, IDF$^+$05] by jointly modeling text features with multinomial distributions and image features with mixture of Gaussian distribution.

In the following experiments, we choose the retrieval models of the Okapi family [RWHB$^+$92] to represent the entire set of probabilistic retrieval models, because they have been proved to effective in text retrieval on a variety of data collections. Besides, Okapi models have been adopted in most of the video retrieval systems currently available [HCC$^+$04, AHI$^+$03, CNL$^+$04, AGC$^+$04, SWG$^+$04, Rea04, KNC05, GSG$^+$03]. However, we also realized that statistical language modeling approaches have recently emerged as a popular probabilistic model for information retrieval due to its elegant statistical foundation and comparable performance with the Okapi family [LZ03, Wes04, CFG$^+$04, SBM05]. But due to space and time limit, we leave the evaluation of language-model based models as future work.

Experiments

To evaluate the performance of various retrieval models on video collections, we designed a series of experiments based on the retrieval models discussed above. We used the last four years of TRECVID corpus as testbeds, i.e., TRECVID'02-'05. For each query topic, the relevance judgment on the search set was officially provided by NIST and the judgment on the development set was collaboratively collected by several human annotators using the Informedia client [HCC$^+$04]. All of the available text sources in the video collections, such as speech transcript, closed caption and video optical character recognition, were indexed to be searchable by the retrieval algorithms. More details of the available text sources can be found in the next section.

In each collection, we have around 25 queries provided by the NIST organizers. For each collection, the text keywords are automatically extracted by extracting the noun phrases from the query description of the original TRECVID queries.

Data	Method	MAP	P30	P100	Person	S-Obj	G-Obj	Sports	Others
t05s	Okapi-SM	0.069	0.174	0.162	0.144	0.058	0.016	0.069	0.017
	Okapi-LM	0.067	0.171	0.160	0.141	0.056	0.016	0.064	0.016
	nnn.ntn	**0.074**	0.172	0.162	0.159	0.057	0.016	0.081	0.016
	lnn.ntn	0.070	0.174	0.161	0.149	0.057	0.016	0.072	0.016
	nnp.ntp	**0.074**	0.174	0.164	0.158	0.057	0.016	0.083	0.016
t04s	Okapi-SM	0.078	0.178	0.105	0.189	0.000	0.039	0.047	0.041
	Okapi-LM	**0.079**	0.180	0.105	0.192	0.000	0.039	0.047	0.042
	nnn.ntn	0.072	0.154	0.100	0.169	0.000	0.039	0.043	0.040
	lnn.ntn	0.078	0.171	0.104	0.186	0.000	0.040	0.046	0.041
	nnp.ntp	0.071	0.152	0.097	0.170	0.000	0.038	0.042	0.035
t03s	Okapi-SM	0.150	0.184	0.120	0.372	0.237	0.067	0.033	0.007
	Okapi-LM	**0.151**	0.187	0.120	0.375	0.245	0.066	0.032	0.008
	nnn.ntn	0.123	0.168	0.111	0.262	0.217	0.065	0.035	0.006
	lnn.ntn	0.142	0.183	0.119	0.339	0.235	0.064	0.037	0.007
	nnp.ntp	0.122	0.168	0.111	0.259	0.219	0.063	0.033	0.006
t02s	Okapi-SM	**0.108**	0.109	0.075	0.175	0.184	0.082	0.000	0.011
	Okapi-LM	0.107	0.113	0.074	0.174	0.184	0.081	0.000	0.011
	nnn.ntn	0.101	0.109	0.073	0.135	0.180	0.085	0.000	0.008
	lnn.ntn	0.101	0.109	0.074	0.134	0.182	0.082	0.000	0.010
	nnp.ntp	0.097	0.101	0.069	0.120	0.180	0.081	0.000	0.009
t05d	Okapi-SM	0.032	0.065	0.058	0.077	0.015	0.006	0.022	0.009
	Okapi-LM	0.031	0.064	0.057	0.073	0.017	0.010	0.021	0.009
	nnn.ntn	**0.036**	0.082	0.061	0.093	0.013	0.004	0.026	0.009
	lnn.ntn	0.033	0.069	0.058	0.082	0.014	0.004	0.023	0.009
	nnp.ntp	**0.036**	0.082	0.062	0.093	0.013	0.004	0.027	0.009
t04d	Okapi-SM	0.073	0.097	0.075	0.130	0.000	0.061	0.077	0.036
	Okapi-LM	**0.074**	0.101	0.075	0.136	0.000	0.068	0.073	0.031
	nnn.ntn	0.065	0.088	0.075	0.116	0.000	0.050	0.063	0.042
	lnn.ntn	0.072	0.092	0.073	0.121	0.000	0.057	0.076	0.045
	nnp.ntp	0.066	0.076	0.075	0.124	0.000	0.050	0.064	0.037
t03d	Okapi-SM	0.092	0.077	0.051	0.185	0.102	0.080	0.048	0.009
	Okapi-LM	**0.095**	0.077	0.053	0.190	0.112	0.082	0.049	0.009
	nnn.ntn	0.078	0.049	0.045	0.172	0.038	0.085	0.047	0.009
	lnn.ntn	0.083	0.069	0.049	0.178	0.065	0.080	0.049	0.009
	nnp.ntp	0.076	0.051	0.046	0.172	0.033	0.084	0.047	0.009

Table 3.2: Comparison of text retrieval models.

Appendix I shows all the text query descriptions and corresponding text keywords extracted from these queries. In order to compare the retrieval performance in a finer granularity, we manually assigned each query into one of the five query classes, that is, named person, named objects, general objects, sports and others [YYH04]. As a preprocessing step, frequent words from a stopword list were removed from both the documents and the queries. Then the Porter stemming algorithm was applied to remove morphological variants. Moreover, because the temporal proximity in the video collection is a strong hint to indicate semantic content closeness [HC04], the retrieval scores were also propagated to a number of nearby documents so as to capture temporal relations within neighbor shots.

As shown in Table 3.2, we implemented five retrieval methods based on both the vector space models and the probabilistic models. Two of them belong to the variants of BM-25 Okapi models which includes the aforementioned SMART version BM-25 model (**Okapi-SM**) and a BM-25 model with different parameters setting $k_1 = 1, b = 0.5$ in order to reduce the effect of document length normalization (**Okapi-LM**). The other three models are based on the vector space models that can be represented using the six-digit SMART codes, i.e., **lnn.ntn** with idf and log-tf weights, **nnp.ntp** with idf weights and pivot length normalization where $s = 0.2$ and p is average document length and **nnn.ntn** with idf weights. For each retrieval method, we reported their mean average precision, precision at 30 documents and precision at 100 documents average over all queries. The mean average precision of each query type are also reported to evaluate the effects of retrieval methods over types of queries.

By comparing these five retrieval approaches, we can observe that both Okapi models outperform the vector space models in almost all the cases with respect to MAP, Prec30 and Prec100, especially on the TRECVID'02-'04 collections. This observation is consistent with previous TREC ad-hoc retrieval evaluation results [RWHB$^+$92] that have demonstrated the effectiveness of Okapi models in the large scale text collections. Moreover, out of the three vector space models, the one using logarithm term frequency and *idf* weights produces a comparable performance with the Okapi retrieval models. In fact, the Okapi models and the *log-tf* model share a common property that the influences of term frequency are both relatively lower in the retrieval function as compared to "raw" term frequency in other models. This shows the usefulness of normalizing the term frequency to a certain range. It also partially explains the popularity of using logarithm *tf* weight in the vector space models. However, there are some exceptions where the Okapi and *log-tf* model does not work well, i.e., the t05d/t05s collections. Unlike other collections, we ran multiple versions of speech recognizers and machine translators on these two multilingual collections (including English, Chinese and Arabic corpus) and thus generated several closely related text sources simultaneously. The raw term frequency (*tf*) turns out to be a more useful factor in this case with high recognition/translation errors but multiple complementary sources. But it is worth pointing out that the differences between these retrieval methods are not statistically significant yet[1]. More experiments are needed to further clarify the advantages and disadvantages between retrieval models.

Along another line, we also notice that the retrieval performance is relatively insensitive to the choice of document length normalization schemes. This is due

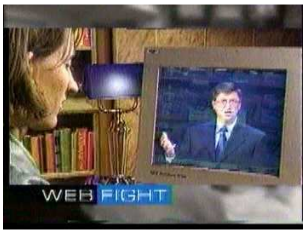
[1] In this paper, we compute the p-value using the sign test and set the significant level to be 1%.

to the fact that the text length in video documents are relatively stable as compared to the regular text documents, which could have very skew length distribution. Finally, the last couple of columns in the table allows us to further analyze the retrieval model behaviors across various query types. Roughly speaking, text retrieval methods are most effective for the queries of finding persons and sometimes for the queries of finding specific objects, because in these cases the users' information needs are defined as terms to be retrieved and thus the truth shots will be shown around associated texts as long as the terms are mentioned in the video. However, the other three types of queries show less benefit from text retrieval, because their information needs are usually less clearly associated text keywords. For example, the information need described in a TRECVID'05 query of "finding a roundtable meeting with a large table" would be much more difficult to captured by the associated text sources than a named person query, "finding Hu Jintao, President of China". For these query types, we should consider incorporating the retrieval results from other modalities to improve retrieval performance.

3.1.2 Text Sources

The text data in video collection is not always generated from a single source alone. Instead, a lot of video corpora such as broadcast news are associated with multiple text sources that can be extracted via manual annotation as well as some well-established automatic techniques such as audio signal processing or visual appearance analysis. Given the many retrieval sources available, it is interesting to study what are distinctive properties for each type of text information and what kinds of text sources can contribute most to the retrieval problem. Generally speaking, the text sources processed in video corpus span several dimensions as follows: [SO03]

- Automatic speech transcripts(ASR) which are converted from raw audio signals by speech recognizers;
- Closed captions(CC) which contain the accurate spoken text written by a person, but usually no time markers for individual words;
- Video optical character recognition(VOCR) extracted from the text visible in the screen;
- Production metadata such as titles, and published descriptions of the video.



VOCR: WeE! Fiighht
ASR: Microsoft the rest of the. Should we be afraid of this computer? is there reason to be great?
CC: Microsoft and the rest of us. Should we be afraid of this computer giant? is there reason to be grateful?

Figure 3.1: Examples of text sources.

Figure 3.1 shows some examples of the text sources mentioned above. Unlike traditional pure text collections, most of the text sources from video are more or less noisy because of either human annotation errors or automatic processing mistakes. Among them, closed captions (if available) are the most complete and accurate source for text information with the lowest word error rate. Unfortunately, closed captions are not always available for retrieval unless the video collections have been manually transcribed or captioned with keywords before the retrieval process. Although a considerable fraction of the television broadcasts have manual transcription nowadays, a much larger number of video collections are unfortunately not transcribed because of the high cost of human transcription [Hau06].

If close captions are not available, speech transcripts are often extracted as an important supplementary text source, which is obtainable through automatic speech recognition [HAH+93, GLA02] and shares a large portion of similar contents with closed captions. Although there are often many recognition errors in speech transcripts, previous experiments [Hau06] had shown as long as speech recognition has a word error rate lower than 35% word error, retrieval performance from the spoken documents is only 3-10% worse than that from the perfect text transcripts. Moreover, at times the speech transcripts can be improved based on evidence from other modalities. For instance, Yang et al.[YCZ+03] attempted to correct non-English names by matching them with VOCR text. It is also worth pointing out that even when the closed captions are available, the retrieval system might still need to consult the corresponding time alignment information from the speech transcripts to synchronize the closed captions with the spoken words. All of these factors have made speech transcripts one of the most indispensable text sources in the video retrieval systems.

Besides closed captions and speech transcripts, another textual features can be derived visually by extracting the overlayed text presented in the video images via video optical character recognition (VOCR) techniques [HCWZ01, SKHS98, CO05]. VOCR is a useful tool to capture people names, event names, location names, as well as product names in commercials that are sometimes not explicitly referred to in the transcript. A complete survey of VOCR related approaches can be found at [Lie03]. VOCR technologies have been commercially available for a long time, but unfortunately the output of VOCR often exhibits a high error rate. For example, the word accuracy of VOCR on the TRECVID'01 video collection is estimated to be as low as 27% [HJN03]. To address this issue, several text correction methods have been proposed to post-process and correct VOCR errors. Among them, dictionary-based correction [HJN03] that expands VOCR words into its other possible spelling based on a dictionary such as MS Word has been demonstrated to be an effective correction approach.

Experiments

In this section, we evaluate five different configurations of the text sources in order to explore the characteristics of each text source for video retrieval. The Okapi-SM retrieval model are used as the baseline retrieval method. Not all the text sources are available in each video collection. In the collection $t02s$, the available sources are speech transcript, production metadata and video OCR. In the other collections, no production metadata is officially provided, but additional closed captions are obtainable in the TRECVID'03-'04 collections. In the following experiments, closed captions and production metadata are officially provided by NIST. The speech transcripts are a mixture of the NIST-provided transcripts and the outputs from a large vocabulary, speaker independent speech recognizer with the word error rate around 30% [HAH^{+}93]. The VOCR transcript is generated by a commercial OCR software which process the filtered images of alphanumeric symbols into text. The screen-overlayed text are further improved through the use of dictionaries and thesauri.

Table 3.3 compares the text retrieval performance on various sources. As expected, if only one text source is chosen, the closed captions (if available) usually provide the best retrieval results due to its low transcription error rate. However, by comparing mean average precision based on speech transcripts and that based on closed captions, we can find that their performance are roughly on par with each other (with a difference around 2% w.r.t. MAP), which indicates the effectiveness of speech transcripts even if they come with a lot of mis-recognition noises. Also, it is not too surprising to observe that VOCR produces the worst MAP among all text sources because the VOCR text has a high recognition error rate and the textual information represented in the screen is often too con-

Data	SRC	MAP	P30	P100	Person	S-Obj	G-Obj	Sports	Others
t05s	A	0.064	0.161	0.158	0.137	0.053	0.015	0.055	0.016
	V	0.029	0.103	0.063	0.053	0.036	0.001	0.044	0.003
	A,V	**0.069**	0.174	0.162	0.144	0.058	0.016	0.069	0.017
t04s	C	0.073	0.175	0.105	0.158	0.000	0.046	0.056	0.038
	A	0.050	0.120	0.080	0.121	0.000	0.022	0.035	0.025
	C,A	0.073	0.167	0.101	0.165	0.000	0.039	0.050	0.044
	V	0.019	0.055	0.022	0.065	0.000	0.002	0.000	0.003
	C,A,V	**0.078**	0.178	0.105	0.189	0.000	0.039	0.047	0.041
t03s	C	0.117	0.176	0.112	0.263	0.176	0.069	0.039	0.007
	A	0.103	0.157	0.106	0.157	0.238	0.060	0.022	0.005
	C,A	0.118	0.177	0.113	0.236	0.211	0.067	0.034	0.007
	V	0.051	0.033	0.018	0.164	0.075	0.007	0.000	0.005
	C,A,V	**0.150**	0.184	0.120	0.372	0.237	0.067	0.033	0.007
t02s	A	0.141	0.135	0.072	0.272	0.177	0.126	0.000	0.009
	P	0.105	0.124	0.080	0.132	0.170	0.102	0.000	0.009
	A,P	**0.141**	0.155	0.088	0.265	0.184	0.128	0.000	0.010
	V	0.002	0.012	0.007	0.002	0.001	0.001	0.000	0.003
	A,P,V	0.108	0.109	0.075	0.175	0.184	0.082	0.000	0.011
t05d	A	0.030	0.067	0.058	0.075	0.009	0.007	0.022	0.009
	V	0.011	0.029	0.018	0.012	0.037	0.000	0.002	0.000
	A,V	**0.032**	0.065	0.058	0.077	0.015	0.006	0.022	0.009
t04d	C	0.071	0.086	0.070	0.120	0.000	0.061	0.068	0.045
	A	0.057	0.093	0.066	0.103	0.000	0.062	0.033	0.027
	C,A	0.074	0.092	0.073	0.125	0.000	0.062	0.077	0.043
	V	0.009	0.037	0.015	0.031	0.000	0.005	0.000	0.000
	C,A,V	**0.074**	0.097	0.075	0.130	0.000	0.061	0.077	0.036
t03d	C	0.069	0.067	0.046	0.083	0.021	0.119	0.050	0.009
	A	0.051	0.059	0.040	0.093	0.028	0.063	0.043	0.006
	C,A	0.068	0.071	0.048	0.110	0.031	0.096	0.048	0.009
	V	0.043	0.027	0.015	0.101	0.079	0.016	0.011	0.000
	C,A,V	**0.092**	0.077	0.051	0.185	0.102	0.080	0.048	0.009

Table 3.3: Comparison of text sources. The second column indicates the available text sources where A means automatic speech transcript, C means closed caption, P means production metadata and V means video OCR.

cise for retrieval purposes. VOCR tends to be more useful for the queries of finding persons than other query types due to the frequent appearance of person names accompanying with their speeches. The production metadata provided in TREC'02 is useful but less effective than using the speech transcript, because they contain less useful semantic content in video-shot levels. Finally, we found that combining two or more different text sources almost always achieves a higher performance than using any of the single sources because they generally contain complementary information. For example, in the collection $t03s$ the configuration that combines all three sources (closed caption, speech transcript and VOCR) together brings a 3.1% MAP improvement over the configuration using closed captions alone. The only exception is in the collection $t02s$, where the performance drops significantly when VOCR adds in. This is because the unexpected low performance of VOCR introduces too much noisy information and thus dilute the better retrieval outputs offered by other sources. When we should combine the results from less accurate sources and when we should not remains an open research topic.

3.1.3 Temporal Expansion Window

In contrast to traditional text collections, video collections have a very distinctive property which might greatly compromise the retrieval performance based on textual features, i.e., the misalignments between relevant video shots and relevant keywords in the corpus. This timing inconsistency between visual appearances and text keywords can partially be explained by the "grammar" in the video production, where the narrative text is designed to introduce or summarize the nearby events shown in a temporal proximity. For example, in a news video footage, an anchorperson or a reporter might summarize the news at the beginning followed by the shots of news events and important persons, resulting in a major time offset between the keywords and the relevant video clips. There are also cases that words may be spoken in the transcript when no associated video clips are present, e.g. a news anchor might discuss a topic for which no video clips are available. This issue will become more serious when we are dealing with the speech transcripts and closed caption rather than dealing with VOCR. Based on the statistics provided by Yang et al. [YCH04], in more than half the cases the relevant shots do not show up in the same shot where the query keywords are mentioned, but before or after the shot.

Generally speaking, there are no simple patterns that can accurately detect such kinds of timing misalignments, but it is arguable that the correct shots are likely to appear in the temporal proximity of the locations where query keywords are mentioned [HC04]. This claim can be confirmed by the successes of extant interactive video browsers that display video shots to users in a temporal order [CM98, ROS04, SWG^{+}04]. Therefore, we can make a legitimate assumption that the closer the shot is to the keyword occurrences, the more possible it has the correct visual appearance. Under this assumption, one common solution to overcome the misalignment problem is to pose an "bell-shape" temporal expansion window on top of the text retrieval results, i.e., add one's text retrieval scores to the nearby shots multiplied by some discount factors α, which is monotonically decreasing with a larger shot-keyword distance. The effectiveness of such a temporal expansion treatment has been demonstrated in many previous studies [YCH04, Rea04, FGJ^{+}04]. In the literature, there exist multiple options for choosing the shape of discount factors, such as using a manual pre-defined windowing function [FGJ^{+}04], a Gaussian distribution function [YCH04], an exponential distribution function [Rea04] and an absolute discount which fixes the discount factor α to the inverse absolute of the shot distance d_s plus 1 ($\alpha_s = 1/|d_s + 1|$).

Experiments

Data	Size	**Limit to Story Boundary**				**No Limits**			
		MAP	P30	P100	R1000	MAP	P30	P100	R1000
t05s	0	0.024(+0%)	0.144	0.103	0.145	0.024(+0%)	0.144	0.103	0.145
	2	0.047(+95%)	0.156	0.143	0.267	0.044(+83%)	0.144	0.133	0.268
	5	0.060(+153%)	0.171	0.154	0.312	0.055(+130%)	0.150	0.142	0.319
	10	0.069(+188%)	0.174	0.162	0.340	0.061(+154%)	0.158	0.149	0.349
	15	**0.072**(+201%)	0.172	0.163	0.353	**0.063**(+161%)	0.158	0.151	0.356
t04s	0	0.055(+0%)	0.141	0.088	0.243	0.055(+0%)	0.141	0.088	0.243
	2	0.071(+29%)	0.152	0.103	0.333	0.071(+29%)	0.154	0.103	0.336
	5	0.076(+39%)	0.170	0.105	0.360	0.077(+41%)	0.172	0.105	0.367
	10	0.078(+43%)	0.178	0.105	0.360	0.079(+45%)	0.180	0.103	0.369
	15	**0.079**(+44%)	0.184	0.107	0.368	**0.080**(+45%)	0.183	0.107	0.362
t03s	0	0.102(+0%)	0.165	0.081	0.269	**0.102**(+0%)	0.165	0.081	0.269
	2	0.135(+32%)	0.181	0.108	0.389	0.091(-10%)	0.148	0.080	0.295
	5	0.144(+41%)	0.185	0.119	0.445	0.096(-5%)	0.137	0.081	0.343
	10	0.150(+47%)	0.184	0.120	0.473	0.099(-2%)	0.141	0.080	0.385
	15	**0.150**(+47%)	0.185	0.120	0.474	0.101(-0%)	0.139	0.083	0.409
t02s	0	0.061(+0%)	0.100	0.055	0.303	0.061(+0%)	0.100	0.055	0.303
	2	0.093(+51%)	0.117	0.076	0.418	0.094(+53%)	0.112	0.072	0.436
	5	0.104(+68%)	0.107	0.076	0.454	0.098(+60%)	0.097	0.075	0.473
	10	**0.108**(+75%)	0.109	0.075	0.475	**0.101**(+63%)	0.087	0.072	0.486
t05d	0	0.012(+0%)	0.051	0.037	0.124	0.012(+0%)	0.051	0.037	0.124
	2	0.021(+82%)	0.063	0.047	0.242	0.020(+67%)	0.060	0.046	0.245
	5	0.028(+137%)	0.067	0.056	0.307	0.025(+113%)	0.058	0.049	0.314
	10	0.032(+170%)	0.065	0.058	0.344	0.027(+131%)	0.060	0.053	0.346
	15	**0.033**(+177%)	0.065	0.058	0.346	**0.027**(+132%)	0.060	0.053	0.329
t04d	0	0.040(+0%)	0.086	0.063	0.311	0.040(+0%)	0.086	0.063	0.311
	2	0.059(+49%)	0.085	0.072	0.429	0.041(+2%)	0.075	0.057	0.366
	5	0.068(+70%)	0.092	0.071	0.479	0.045(+12%)	0.076	0.056	0.413
	10	**0.073**(+83%)	0.097	0.075	0.487	0.048(+21%)	0.079	0.058	0.434
	15	**0.073**(+82%)	0.100	0.075	0.521	**0.049**(+23%)	0.083	0.058	0.459
t03d	0	0.070(+0%)	0.077	0.040	0.258	0.070(+0%)	0.077	0.040	0.258
	2	0.085(+21%)	0.080	0.047	0.377	0.084(+19%)	0.079	0.047	0.373
	5	0.087(+23%)	0.080	0.050	0.458	0.084(+19%)	0.077	0.049	0.425
	10	**0.092**(+30%)	0.077	0.051	0.480	**0.089**(+26%)	0.079	0.052	0.463
	15	**0.092**(+30%)	0.076	0.051	0.481	**0.089**(+26%)	0.076	0.052	0.469

Table 3.4: Comparison of expansion window sizes.

Two factors are needed to decide in a temporal expansion process, i.e., the form of discount factors and the temporal expansion window size. But to avoid explosive combinations of experimental configurations, we specifically adopt the absolute discounting schemes in the following discussions and leave the evaluations of other discount factors to future work. Therefore we only have to set the expansion window size that controls how many shots before and after the retrieved shots are expanded by adding the additional discounted retrieval scores.

The effect of varying expansion window size is compared in Table 3.4. The following window sizes have been chosen in our experiments, i.e., 0,1,2,5,10,15. The third column in Table 3.4 shows that the retrieval performance in terms of mean average precision can always become higher with a larger expansion size and the performance improvement is statistically significant. The improvement is much more considerable when the expansion window size is less than 5, while the performance is gradually saturated to a asymptotic level afterwards. Not surprisingly, the major performance growth factor for a larger window size can be traced to a higher overall recall rate brought by more shots expanded (e.g., the recall at 1000 shots in t05s grows from 14.5% to 31.2% with an expansion size of 5 shots), while the change on the overall precision is not as obvious as recall in these collections.

Given that most of our collections are news video archives, we also compare the following two settings: allow the shots outsides the same story boundaries to be expanded (**No Limit**) and not allow them to be expanded (**Limit to Story Boundary**) [2]. This comparison could give us some evidence if the misalignments between relevant video clips and relevant keywords would go beyond news story boundaries. As shown in the Table 3.4, limiting the expansion within story boundaries **"Limit to Story Boundary"** almost always produces better MAP than the other case **"No Limit"**. Even worse, sometimes expanding more video shots in the setting of **"No Limit"** might compromise the retrieval results such as in collection $t03s$. This series of experiments suggest that we should not expand retrieval results to the video shots outside the story where the relevant keywords are found.

3.1.4 Query Expansion

Most users begin their retrieval process without knowing the detailed collection information and the retrieval environment. At first, they usually find it difficult to formulate a query that well satisfies the information need, and then they will

[2] A news story is defined as a segment of a news broadcast with a coherent news focus which contains at least two independent, declarative clauses [SO03]

refine the queries again and again until the retrieval purpose is reached. This suggests us to consider an iterative query refinement process that can re-construct the query representation in the hope of retrieving additional useful documents. Two basic approaches for such an query reformulation are available, i.e., query expansion that expand new terms to the original queries and term reweighting that modified the query term weights [BYRN99]. Due to the problem of short document length and high transcription errors in the text sources of video collections, query reformulation becomes more useful and critical as they can capture the exact information needs and introduce additional retrieval information from users or external sources. In this section, we particularly focus on examining the effects of query expansion methods, which can be grouped into three categories as follows [BYRN99]:

1. Methods based on manual adjustment or relevance feedback from users,

2. Methods based on the set of documents initially retrieved,

3. Methods based on global information from the entire document collection.

The most straightforward query expansion approach is to directly ask users to modify the queries in each iteration. Users can provide additional keywords or substitute previous keywords after reviewing initial retrieved documents. However, this process usually requires an intensive efforts from users to come up with appropriate modifications. A more user-friendly approach is to utilize the users' relevance feedback to update the query words. It begins with asking users to label the relevant documents from an initial list of documents, extract terms from the relevant documents provided by users and finally append the additional terms to the query. This is the best choice if users are willing to provide additional labelings. One of the earliest relevance feedback algorithms was proposed by J.J. Rocchio [Roc71]. The feedback iterations modify the query vectors by iteratively increasing the weights of terms contained in positive documents and penalizing the terms in negative documents. Many extensions such as Ide regular algorithm and Ide dec-hi algorithm [Ide69] are proposed based on Rocchio algorithm. The recent advancements of machine learning introduced new alternatives for the relevance feedback algorithms. Support vector machines(SVMs) [DSG01] has been applied for relevance feedback and achieved much better performance than the Rocchio algorithm especially at the beginning of the feedback iterations. Instead of the explicit feedback, White et al. [WJR05] consider the form of implicit feedback which monitors searcher interaction with different representations of top-ranked documents and chooses new retrieval strategies accordingly.

Unfortunately, relevance feedback requires input from additional human interactions. Sometimes it is more reasonable to obtain the expanded queries in an automatic manner. To achieve this, we can adopt either a local strategy that explores the information from initially retrieved documents or a global strategy that analyzes document statistics based on the global collections. For the local strategy, initial retrieved documents are examined in real time to determine the terms to expand. The essence of local strategies is to utilize top retrieved documents as positive example to select expanded discriminative query terms to improve the retrieval performance. In these approaches, a small number of top-ranked documents are assumed to be relevant and considered in a automatic feedback process to modify the queries. Sometimes, this types of algorithms are also called pseudo-relevance feedback(PRF). One such algorithm is called local context analysis(LCA) proposed by Xu et al. [XC00]. In this work, several noun groups are selected from the top ranked documents based on the passage co-occurrence of query terms and introduced into the original query. More details of LCA can be found at [XC00].

In contrast to the local strategy, the global strategy is to expand the query description using information from the whole collection or external thesaurus. For example, a global analysis method is to leverage external knowledge sources such as a co-occurrence thesaurus or semantic network. An example of semantic network is WordNet [Fel98], an online lexical reference system whose design is inspired by the current psycholinguistic theories of human lexical memory. Based on these thesauri or semantic networks, we can introduce related terms according to their relationship to the query terms. For global analysis, another approach is to automatically create a domain-specific thesaurus from the text collection based on global term-to-term similarities [QF93]. A decade ago, it was believed that global analysis cannot generate consistent retrieval performance improvement over general collection, but recently this belief has been changed with the appearance of modern global analysis approaches. Therefore it will be very interesting to evaluate how these methods perform on the context of video collections.

A number of researchers in the video retrieval community began to investigate various query expansion techniques. For instance, Chua et al. [CNL$^+$04] evaluated the pseudo relevance feedback technique in the TRECVID'04 dataset, which uses top retrieved documents to obtain a list of additional query keywords and iterate the retrieval process. Their results showed that pseudo relevance feedback can bring a small but not significant improvement over the non-feedback baseline. Kennedy et al. [KNC05] augmented the text retrieval results via two global analysis approaches, i.e., leveraging external knowledge sources of WordNet and

Google to enrich the query representation. However, their work did not explicitly report the text retrieval performance after query expansion. To expand text query to account for high-level semantic concepts, Neo et al. [NZKC06] make use of the WordNet hierarchy and Resnik information-content metric to estimate a heuristic combination weight for semantic concepts. It brings an additional 0.4% MAP improvement over the direct keyword matching approach. A more recent study [VN06] compared three automatic query expansion techniques including Rocchio-based query expansion, lexical-context based expansion and sematic annotation-based expansion on the TRECVID datasets. Surprisingly, their experiments have underscored the difficulty of automatic query expansion in video collections, because only one out of three approaches can gain higher average precision than non-expansion baseline. However, they also suggested that combining both the retrieval results without and with query expansion can produce better results than either one of them, especially when the combination is carried out in a query-dependent manner. To summarize, query expansion has shown its potentials, but further experiments are still necessary to prove that query expansion is able to significantly improve the text retrieval performance in video collections.

Experiments

To evaluate the effectiveness of query expansion in the video corpus, we designed and evaluated three types of expansion approaches. The first approach is manual expansion (**Manual**) which asks users to manually introduce additional query words and refine the text queries based on the information of development data. The second method is based on a global query expansion strategy(**WordNet**). It passes every query keywords into WordNet and expands a fixed number of synonyms to the original queries. The last method is based on a local query expansion strategy by analyzing the relevance of top retrieved documents(**Local**). The expanded terms are chosen to be the terms with the highest $tf.idf$ features in the top 10 retrieved documents.

Table 3.5 compares all three expansion methods and the baseline method on TRECVID'03-'05 collections. The second column of Table 3.5 indicates the labels of methods and corresponding expansion parameters, where the parameter following "WordNet" indicates the number of synonyms expanded and the number following "Local" indicates the number of query terms expanded. The message from the experimental results is mixed: manual expansion can considerably boost the retrieval results which shows the usefulness of leveraging additional human knowledge, but the other two types of automatic expansion approaches is not

Data	App.+Para.	MAP	P30	P100	R1000	Person	S-Obj	G-Obj	Sports	Others
t05s	Baseline	0.069(+0%)	0.174	0.162	0.340	0.144	0.058	0.016	0.069	0.017
	Manual	**0.103(+49%)**	0.226	0.200	0.383	0.194	0.086	0.013	0.171	0.021
	WordNet 1	0.060(-13%)	0.136	0.138	0.341	0.133	0.052	0.014	0.046	0.011
	WordNet 2	0.059(-14%)	0.135	0.135	0.340	0.133	0.051	0.014	0.046	0.010
	Local 1	0.070(+1%)	0.172	0.163	0.341	0.142	0.057	0.015	0.077	0.017
	Local 2	0.067(-3%)	0.169	0.158	0.341	0.140	0.055	0.014	0.069	0.017
	Local 3	0.065(-6%)	0.165	0.155	0.338	0.136	0.055	0.013	0.062	0.017
t04s	Baseline	0.078(+0%)	0.178	0.105	0.360	0.189	0.000	0.039	0.047	0.041
	Manual	**0.093(+18%)**	0.180	0.118	0.459	0.200	0.000	0.030	0.156	0.044
	WordNet 1	0.052(-33%)	0.103	0.071	0.302	0.104	0.000	0.037	0.027	0.035
	WordNet 2	0.048(-39%)	0.090	0.068	0.296	0.095	0.000	0.035	0.037	0.027
	Local 1	0.071(-9%)	0.165	0.103	0.357	0.169	0.000	0.031	0.047	0.041
	Local 2	0.063(-19%)	0.149	0.102	0.358	0.142	0.000	0.029	0.045	0.041
	Local 3	0.060(-23%)	0.145	0.101	0.358	0.132	0.000	0.027	0.047	0.041
t03s	Baseline	0.150(+0%)	0.184	0.120	0.473	0.372	0.237	0.067	0.033	0.007
	Manual	**0.194(+29%)**	0.227	0.155	0.561	0.404	0.299	0.099	0.135	0.041
	WordNet 1	0.135(-9%)	0.168	0.114	0.490	0.298	0.214	0.081	0.036	0.007
	WordNet 2	0.119(-20%)	0.145	0.098	0.463	0.265	0.185	0.071	0.035	0.006
	Local 1	0.150(+0%)	0.180	0.119	0.473	0.333	0.243	0.066	0.032	0.007
	Local 2	0.143(-4%)	0.180	0.118	0.479	0.351	0.243	0.057	0.032	0.007
	Local 3	0.141(-5%)	0.165	0.116	0.488	0.364	0.235	0.051	0.025	0.007
t05d	Baseline	0.032(+0%)	0.065	0.058	0.344	0.077	0.015	0.006	0.022	0.009
	Manual	**0.056(+75%)**	0.090	0.089	0.407	0.089	0.017	0.034	0.153	0.011
	WordNet 1	0.028(-10%)	0.054	0.052	0.331	0.075	0.010	0.005	0.015	0.008
	WordNet 2	0.028(-12%)	0.053	0.052	0.332	0.075	0.010	0.002	0.015	0.006
t04d	Baseline	0.073(+0%)	0.097	0.075	0.487	0.130	0.000	0.061	0.077	0.036
	Manual	**0.094(+28%)**	0.115	0.084	0.691	0.174	0.000	0.079	0.113	0.031
	WordNet 1	0.056(-23%)	0.074	0.049	0.440	0.065	0.000	0.067	0.069	0.033
	WordNet 2	0.049(-33%)	0.075	0.045	0.417	0.057	0.000	0.064	0.046	0.032
t03d	Baseline	0.092(+0%)	0.077	0.051	0.480	0.185	0.102	0.080	0.048	0.009
	Manual	**0.116(+26%)**	0.092	0.066	0.579	0.199	0.144	0.109	0.084	0.012
	WordNet 1	0.081(-11%)	0.065	0.044	0.508	0.186	0.055	0.079	0.041	0.008
	WordNet 2	0.068(-25%)	0.043	0.036	0.468	0.167	0.030	0.067	0.040	0.008

Table 3.5: Comparison of query expansion methods.

so consistent in producing better performance in terms of average precision, especially when the number of expansion terms grows larger. The inconsistency of the last two approaches can be traced back to the noticeable degradation in precision (even though recall is slightly higher than before). This is because many additional "noisy" terms are introduced into the query after the step of query expansion. Another reason for their inconsistency can be attributed to the subpar retrieval performance in video collections, which prevents automatic query expansion techniques from significantly improving the search results by assuming top-ranked documents are mostly relevant. To further compare their performance w.r.t each query types, it can be found that the finding-person queries degrade the most with the automatic expansion, which suggests query expansion is not effective to search for person-related shots. Interestingly, the finding-sport queries gains a large amount of benefits from manual query expansion, indicating the potential of expanding extra sports-related words in query topics. To summarize, the best query expansion approach for video corpus so far is to expand keywords in a manually controlled manner rather than in an automatic way.

Note that, what we attempt to emphasize is the difficulty rather than the failure of using automatic query expansion to improve video retrieval performance. In fact, our experiments can be designed and implemented more carefully in several aspects. For example, following the observation that query expansion could potentially be helpful for sports queries but less useful for named person queries, we can tune the expansion methods and combine them with original retrieval results based on the information of individual query types. It is also possible to introduce additional hyponyms of the query words from the WordNet other than the synonyms. All these variations in automatic query expansion can be explored in the future.

3.2 Image Retrieval

Content-based image retrieval(CBIR) techniques have been developed for more than one decade [SWS+00]. Its goal is to search a given image collection for a set of relevant images that are similar to the query images. Previous research efforts have led to many successful image retrieval systems such as MARS [RHM97], VisualSeek [SC96c], QBIC [FBF+94], SIMPLicity [LWW00] and so on. For the task of video retrieval, although CBIR is not so powerful as text retrieval in terms of handling general semantic queries, it is still useful in dealing with a number of queries from several specific domains, where the information needs have to be consistent with visual appearances. For instance, CBIR has great success when the retrieval targets are related to sport events or finding duplicate commercial shots.

Typical CBIR systems are built based on a vector space model which represents an image as a set of features and the difference between two images is measured through a similarity function between their feature vectors. They start with a few image examples as inputs, convert them into sets of image features, match them with the features of all images in the entire collection, and retrieved the closest ones. In the rest of this section, we discuss each individual component of image retrieval systems and evaluate them in the context of video collections.

3.2.1 Image Features

Similar to the term weights in text retrieval, image features are represented as a vector of real values, which aims to compress high-dimensional image information into a lower dimensional vector space and thus make it indexable and comparable with other images. In the literature, there are mainly three types of (low-level) image features that have been applied, i.e., color-based features, texture based features, and shape-based features [AKJ02, SWS$^+$00, RHC97][3].

Color-based features have been shown to be the most widely-used features in CBIR systems and have also been demonstrated to be the most effective features in the TRECVID evaluation [Rea04, AHI$^+$03, HBC$^+$03, FGJ$^+$04, CFG$^+$04, AGC$^+$04]. This is because color features maintain strong cues that capture human perception in a low dimensional space and they can be generated with less computational efforts than other advanced features. Most of them are independent of variations of view and resolution, and thus possess the power to locate the target images robustly. Many color spaces have been suggested in previous studies such as RGB, YUV, HSV, HVC, L*u*v*, L*a*b* and the Munsell space [Bim01]. The simplest representation of color-based features is the color histogram, where each component in the color histogram is the percentage of pixels that are most similar to the represented color in the underlying color space [FBF$^+$94, SC96c]. Another type of color-based image feature is called color moments which only compute the first two or three central moments of color distributions in the image [SO95, SC96b] with other information discarded. They aim to create a compact and effective representation in image retrieval. Other than these independent color representations, Pass et al. [PZ99] developed color coherence histograms (CCHs) to overcome the matching problems of standard color histograms. It includes both the spatial information along with the color density information in a single representation. Huang et al. [HKM$^+$97] proposed the use of color correlogram over the CCHs. A color correlogram expresses the spatial correlation of pairs of colors with distance information, thus making it robust against the change of viewpoint, zoom-in and zoom-out, etc.

Texture-based features aim to capture the visual characteristics of homogeneous regions which do not come from the presence of a single color or intensity [SC96a]. These regions may have unique visual patterns or spatial arrange-

[3] There are many other approaches to produce image features for content-based image retrieval such as salient points and so on, but it is not our focus to provide an exhaustive list for all of them. A more complete survey of CBIR can be found at Antani et al. [AKJ02] and Smeulder et al. [SWS$^+$00].

ments of pixels which gray level or color features in a region may not sufficiently describe. The process of extracting the texture-based features often begins with passing the images into a number of Gabor or Haar wavelet filters [Lee96, AHI^{+}03] based on the assumption of existing locally homogeneous regions. After the filtering process, the feature vector can be either constructed by concatenating the central moments from multiple scales and orientations into a long vector [MM96, NPZ01, SC96a] or generated by the image distribution directly [PHB97, TNP96]. In the literature, there are a few review papers that aim to investigate the effectiveness of various types of texture features. For instance, [PPO92] compared four types of texture representations and observed that the co-occurrence matrix representation work the best in their test collections. [MM95] evaluated the wavelet texture features for the task of image annotation, which includes orthogonal/bi-orthogonal wavelet transform, tree-structured wavelet transform and Gabor wavelet transform. They concluded that Gabor wavelet representation was the best among all the tested features.

To capture the information of object shapes, a huge variety of shape-based features have been proposed and evaluated [ZR72, CK96, LM94, MKL97]. Shape features can be either generated from boundary-based approaches that use only the outer contour of the shape segments, or generated from region-based approaches that considers the entire shape regions [RSH96]. One of the simplest approaches to extract shape features is to detect visible edges in query images and then match their edge distribution or histogram against those of the target images [MH79, HCC^{+}04, CFG^{+}04]. Some advanced methods adopt the deformable image templates to match user sketch in the target images [BP97]. Since the user sketch may not exactly match with the shapes in the collection, these methods have to elastically deform the user template to match the image contour. Another approach to extract shape features is to use implicit polynomials for effective representation of the geometric shape structures [LTC97], which is robust, stable to the general image transformation. [MKL97] presented a comprehensive comparison of shape features for retrieval by evaluating them on a 500-element trademark dataset. Another review papers about shape-based features can be found at [LM94].

It is not always necessary to construct the image features by globally extracting features from the entire image. Although global image features are efficient to compute and provide reasonable retrieval capabilities, they are very likely to generate unpredictable false positive due to its concise representations [RHC97]. In fact, these features can be extracted from a finer granularity, such as regular image grids/layouts, automatically segmented image blobs and local feature points.

In practice, content-based image classification/retrieval based on regional features usually shows better performance than its counterpart using global features, although this might lead to a higher computational cost in the step of feature extraction. In the following discussions, we review some general methods on extracting local image features.

To derive local features from images, a natural idea is to partition the entire image into a set of regular image grids and extract image features (especially color features) from image grids [AHI$^+$03, FBF$^+$94, CTO97]. For instance, Cooke et al. [CFG$^+$04] used a local color descriptor based on the average color components on an 8x8 block partition of images. Hauptmann et al. [HCC$^+$04] studied color layout features on a 5x5 regular image grid. A 4x4 spatial image grid is used in [AGC$^+$04]. Extended from regular image grids, quad-tree based layout [LOT94] approaches first split the image into a quad-tree structure and construct color histogram for each tree branch so as to describe its local image content. Although being simple in their intuitions, concepts and implementations, regular-image-grid based approaches could be still too coarse to provide sufficient local information for the retrieval task. Therefore several other image layout representations have been proposed before. For instance, Stricker and Dimai [SO95] predefined five partially overlapped regions and extracted the first three color moments from each region, where the advantage of the overlapping regions is their relative insensitivity to small regional transformations. The representation of color tuple histogram is suggested in [RS96], which first builds a codebook to represent every combination of coarsely quantized close hues and then compute a local histogram based on the quantized hues. The color coherent histogram [PZ99] and color correlogram [HKM$^+$97] are two more examples of advanced image representations that take image spatial information into account.

In order to locate specific objects in images, it would be advantageous to extract image features (e.g., color features or shape features) from segmented image regions. Image segmentation is defined as "a division of the image data into regions in such a way that one region contains the pixels of the silhouette of the given object without anything else." [SWS$^+$00]. This task is so important in the literature of compute vision that a huge variety of segmentation approaches have been proposed before. Survey of mainly historical interests can be found at [Nev86, MA85, PP93]. Most segmentation algorithms proceed by automatically clustering the pixels into groups. A number of graphical theoretical clustering approaches have been proposed before [SB96, CRZ96] due to their ability to deal with any affinity function. For example, the normalized cut algorithm proposed by Shi and Malik [SM98, SM00] have been widely applied in the task of visual retrieval, object recognition and motion segmentation. Numerous alternative criteria have also been suggested for segmentation [PF98, CRZ96]. The usage of image segments have been widely studied in the context of content-based video retrieval, such as [SBM05, ZLC$^+$05, HC04]. Note that, in this case, the requirement of segmentation accuracy is highly dependent on the choice of image features. For the color features, a coarse segmentation should be sufficient, while for the shape features, accurate segmentation is usually desirable. Last comment worth mentioning for segmentation is that segmentation for general objects in broad domains is unlikely to succeed, although there are some exceptions for some sophisticated methods in the narrow domains [SWS$^+$00].

One way to circumvent the brittleness of segmentation but maintain the local information of images is to extract image features from selected salient points (a.k.a. feature points). Since the image information is concisely summarized by a limited number of salient points, these points should be selected with highest saliency and robustness. In [CBGM97], a mixture of Gaussian model is estimated to model the distribution of salient points into feature space. The information of the homogeneous regions is captured by the means and covariances of the Gaussian components. To improve the quality of feature description, invariant and salient features of local patches have also been considered [TG99]. In [SM97], salient and invariant transitions are recorded in gray images. To localizes all the occurrences of a query object in videos, "Video Google" [SZ03] represents the objects by a set of SIFT-based viewpoint invariant descriptors, and thus recognition are work robustly to changes in viewpoint, illumination and partial occlusion. Chang et al. [CHK$^+$05] investigated a part-based object representation in TRECVID collections in order to capture the spatial relationship and the local attributed of salient parts. Zhai et al. [ZLC$^+$05] also evaluated the performance of the local features of image segments and feature points in the video collections.

Experiments

To evaluate the performance of image retrieval over the TRECVID corpus, we have extracted three types of low-level features as described above, i.e., color based features, texture based features and edge based features. By default, image features are generated over 5x5 regular grids posed on every image. All grid features are concatenated into a longer vector as the features of the entire image unless stated otherwise. Each dimension of the feature vectors is normalized by its own variance. Finally, we compute a harmonic mean of the Euclidean distances from each image example to the document keyframe as the image retrieval output.

Data	Feature	Para.	MAP	P30	P100	Person	S-Obj	G-Obj	Sports	Others
t04s	color:hsv	his:5x5	0.004	0.010	0.007	0.000	0.000	0.002	0.024	0.000
	color:hsv	mom:5x5	0.014	0.045	0.028	0.003	0.000	0.001	0.095	0.003
	color:rgb	his:5x5	0.008	0.029	0.018	0.004	0.000	0.001	0.055	0.000
	color:rgb	mom:5x5	**0.016**	0.048	0.032	0.022	0.000	0.001	0.075	0.001
	texture	mom:3x3	0.003	0.010	0.010	0.001	0.000	0.000	0.017	0.001
	texture	mom:5x5	0.003	0.006	0.010	0.001	0.000	0.000	0.019	0.000
	edge	his:5x5	0.003	0.013	0.011	0.000	0.000	0.002	0.016	0.001
t03s	color:hsv	his:5x5	0.026	0.072	0.046	0.000	0.078	0.001	0.113	0.008
	color:hsv	mom:5x5	0.035	0.087	0.058	0.000	0.064	0.010	0.221	0.010
	color:rgb	his:5x5	0.029	0.043	0.030	0.000	0.114	0.001	0.065	0.002
	color:rgb	mom:5x5	**0.049**	0.088	0.060	0.000	0.095	0.010	0.313	0.010
	texture	mom:3x3	0.024	0.036	0.026	0.000	0.052	0.002	0.161	0.000
	texture	mom:5x5	0.016	0.032	0.027	0.000	0.005	0.004	0.169	0.000
	edge	his:5x5	0.028	0.056	0.036	0.001	0.065	0.013	0.124	0.002
t04d	color:hsv	his:5x5	0.016	0.036	0.020	0.023	0.000	0.007	0.051	0.000
	color:hsv	mom:5x5	0.018	0.051	0.023	0.029	0.000	0.007	0.050	0.001
	color:rgb	his:5x5	**0.021**	0.057	0.025	0.033	0.000	0.005	0.065	0.002
	color:rgb	mom:5x5	0.016	0.050	0.025	0.034	0.000	0.001	0.039	0.005
	texture	mom:3x3	0.001	0.004	0.003	0.000	0.000	0.000	0.007	0.000
	texture	mom:5x5	0.001	0.011	0.005	0.004	0.000	0.000	0.001	0.000
	edge	his:5x5	0.016	0.051	0.024	0.035	0.000	0.000	0.034	0.005
t03d	color:hsv	his:5x5	0.031	0.039	0.018	0.006	0.130	0.004	0.027	0.001
	color:hsv	mom:5x5	0.034	0.035	0.022	0.006	0.131	0.005	0.053	0.002
	color:rgb	his:5x5	0.038	0.039	0.021	0.006	0.153	0.008	0.032	0.002
	color:rgb	mom:5x5	**0.044**	0.053	0.026	0.006	0.153	0.012	0.088	0.006
	texture	mom:3x3	0.006	0.011	0.005	0.000	0.005	0.000	0.064	0.000
	texture	mom:5x5	0.005	0.011	0.004	0.000	0.001	0.002	0.058	0.000
	edge	his:5x5	0.029	0.044	0.017	0.000	0.070	0.019	0.109	0.000
t02s	color:hsv	mom:5x5	**0.029**	0.057	0.033	0.127	0.012	0.008	0.000	0.015
	texture	mom:5x5	0.006	0.021	0.016	0.009	0.003	0.006	0.000	0.007

Table 3.6: Comparison of image features(I).

The details of the feature generation process are shown as follows,

- The color features are computed based on two different color spaces including the HSV space and the RGB space. For each color space and image grid, we extract both a full color histogram with a 5-bin quantization of every color channel and a color moment histogram including the first and second moments.

- The texture features are obtained from the convolution of the image pixels with various Gabor wavelet filters. For each filter we computed a histogram which was quantized into 16 bins. Their central and second-order moments are concatenated into a texture feature vector. Two versions of texture features are derived in our implementation: one uses 6 filters in a 3x3 image grids and the other uses 12 filters for 5x5 image grids.

Data	Feature	Para.	MAP	P30	P100	Person	S-Obj	G-Obj	Sports	Others
t05s	color:hsv	his:5x5	**0.039**	0.114	0.078	0.033	0.021	0.003	0.186	0.005
	color:hsv	mom:5x5	0.032	0.090	0.057	0.007	0.021	0.003	0.194	0.004
	color:rgb	his:5x5	0.037	0.114	0.081	0.019	0.022	0.005	0.202	0.004
	color:rgb	mom:5x5	0.034	0.101	0.056	0.004	0.022	0.004	0.211	0.003
	texture	mom:3x3	0.005	0.026	0.019	0.001	0.006	0.001	0.028	0.001
	texture	mom:5x5	0.003	0.024	0.021	0.000	0.006	0.002	0.009	0.001
	edge	his:5x5	0.009	0.042	0.035	0.000	0.014	0.003	0.044	0.003
t05d	color:hsv	his:5x5	0.060	0.144	0.075	0.028	0.082	0.003	0.260	0.009
	color:hsv	mom:5x5	0.058	0.125	0.075	0.007	0.072	0.007	0.309	0.004
	color:rgb	his:5x5	**0.097**	0.160	0.090	0.028	0.127	0.030	0.451	0.010
	color:rgb	mom:5x5	0.076	0.135	0.074	0.017	0.097	0.010	0.385	0.005
	texture	mom:3x3	0.029	0.050	0.028	0.004	0.097	0.000	0.051	0.003
	texture	mom:5x5	0.026	0.051	0.025	0.003	0.059	0.000	0.100	0.000
	edge	his:5x5	0.029	0.075	0.035	0.010	0.038	0.001	0.146	0.001

Table 3.7: Comparison of image features(II).

- The edge histogram are summarized from the outputs of a Canny edge detector. It includes a total of 73 bins, where the first 72 bins represent the edge directions quantized at a 5 degree interval and the last bin represents a count of the number of pixels that are not contributed to any edges.

The comparison of various image features are shown in Table 3.7. As shown in the experiments, image retrieval can only achieve a poor 2% - 3% mean average precision for almost all the collections. Obviously it is not as effective as the text retrieval on average. As a special case, image retrieval works relatively better in the latest TREC'05 collection, mainly due to the superior effectiveness of image similarity matching in the sports queries. Among all kinds of color features, the color moment features in both the HSV/RGB space have the best performance on average, followed by the features of color histogram. This again confirms the effectiveness and explains the popularity of color-based features. Occasionally, the edge histogram can provide a comparable performance with color features, but its performance is not as consistent as color features across all the collections. The texture features, unfortunately, are among the worst features in all the collection due to their inability to capture the semantics in general non-texture images. Taking a deeper look at the performance distribution on each query type, we found that the best candidates for image retrieval are the queries of finding specific objects and sport events. This is understandable because target specific objects and sport events usually share consistent visual appearances with the given image examples. But image retrieval does not seem to be a good idea for the other three types of queries, i.e., person-finding, object-finding and other general queries.

3.2.2 Distance Metric

Image retrieval algorithms usually sort and retrieve the relevant images based on a predefined similarity measure (distance metric) between query examples and indexed images. Previous studies show that the choice of similarity measures is critical to the image retrieval performance [AKJ02]. Thus, a large number of distance metrics have been proposed and tested in the literature. In the following discussion, we discuss several common distance metrics under the assumption that only one query image is available.

Let $f_q(i)$ denote the i^{th} feature of the query example and $f_c(i)$ denote the i^{th} feature of the indexed image to be compared, where $i = 1..N$. Two widely used distance metrics, i.e., Euclidean distance (L_2 distance) and absolute distance (L_1 distance), are both the special cases of the L_M metric or Minkowski distance metric defined in Eqn(3.4). An extension of the Euclidean distance is the Mahalanobis distance presented in Eqn(3.5), where the inverse of a covariance matrix C is plugged into the quadratic function to associate different weights with each feature dimension.

If the underlying features are computed in the form of histograms, the retrieval system usually adopts some distance metrics that capture the difference between two probability distributions. For example, we can compute the simple histogram difference as the absolute differences of the feature histograms. As a more reliable distance metric w.r.t. histogram features, the color histogram intersection as shown Eqn(3.6) in was proposed for the image retrieval task [SB91] where a value close to 1 indicates a high similarity. Eqn(3.7) shows the χ^2 distance for comparing two histograms proposed by Nagasaka et al. [NT92] where the low value indicates a good match. Its underlying idea is to find the images with histogram distributions least independent to the query examples. Stricker [Str94] has studied the discrimination ability of histogram-based indexing methods. He concluded in his work that the histogram-based technique would only work effectively when the histograms are sparse. Beyond using the fixed distance metrics, numerous relevance feedback and manifold learning approaches [RHM97, HLZ$^+$04, HMK$^+$02, SLZ01] have also been proposed to learn the distance metrics adaptively based on the information of user feedback.

$$D_{LM} = \left(\sum_{i=1}^{N} |f_q(i) - f_c(i)|^M \right)^{1/M} \tag{3.4}$$

$$D_{MB} = (\vec{f_q} - \vec{f_c})^T C^{-1} (\vec{f_q} - \vec{f_c}) \tag{3.5}$$

$$D_{HI} = \frac{\sum_{i=1}^{N} \min(f_q(i) - f_c(i))}{\sum_{i=1}^{N} f_q(i)} \tag{3.6}$$

$$D_{CHI} = \sum_{i=1}^{N} \frac{(f_q(i) - f_c(i))^2}{f_q(i) + f_c(i)} \tag{3.7}$$

Most image retrieval algorithms simply consider dealing with one query example at a time. But since it is not impossible that users could simultaneously provide multiple image examples to the retrieval systems, we might need to come up with some approaches to aggregate all of the distance metrics from each query image to be a final ranked list. The common approach is to measure image similarities from individual query images and then fuse the similarity measures into a single output via certain kinds of fusion methods. We will consider five types of common aggregation functions in our following experiments, i.e., maximum, minimum, harmonic mean, average(arithmetic mean) and product(geometric mean) [AHI$^+$03, HBC$^+$03]. Several advanced multi-query-example retrieval approaches have also been proposed before. McDonald et al. [MS05] studied the effect of various combination strategies for merging multiple visual examples. Jin et al. [JH02] proposed a probabilistic image retrieval model by computing the conditional probability of generating the target image given multiple query images. Westerveld et al. [WdV04] developed a document generation model to handle multi-example queries, which capture all the information available in the query examples with a limited number of Gaussian components. Natsev et al. [NS03] considered three types of criteria to automatically select the most effective image examples for retrieval, i.e., KMEANS which uses the mean of image clusters as queries, MINDIST which finds the most distinct positive examples in a greedy way, and SUMDIST which provides a compromise between KMEANS and MINDIST criteria.

Experiments

We compared three types of distance metrics in Table 3.8 including the L_2, L_1 and χ^2 metrics. The underlying image features are chosen to be color moments on the HSV color space. Each dimension is normalized by its own variance. From Table 3.8, we observe that using the L_1 distance metric usually work slightly better than using the L_2 and χ^2 distance, but their differences are not statistically significant. Therefore, it is not conclusive yet to judge which metric is the best choice for image retrieval. But being robust to outliers and efficient to compute [LR87], the L_1 distance seems to be one of the most effective metrics in practice.

Data	Dist.	MAP	P30	P100	Person	S-Obj	G-Obj	Sports	Others
t05s	L_2	0.025	0.076	0.048	0.010	0.021	0.002	0.130	0.003
	L_1	**0.031**	0.089	0.060	0.008	0.022	0.003	0.183	0.004
	χ^2	0.025	0.085	0.055	0.007	0.014	0.003	0.146	0.003
t04s	L_2	0.014	0.045	0.028	0.003	0.000	0.001	0.095	0.003
	L_1	**0.017**	0.046	0.032	0.007	0.000	0.002	0.107	0.002
	χ^2	0.015	0.043	0.031	0.002	0.000	0.001	0.097	0.002
t03s	L_2	0.035	0.087	0.058	0.000	0.064	0.010	0.221	0.010
	L_1	**0.039**	0.085	0.055	0.000	0.079	0.007	0.238	0.009
	χ^2	0.037	0.079	0.056	0.001	0.076	0.010	0.210	0.010
t05d	L_2	0.044	0.096	0.061	0.006	0.066	0.003	0.216	0.004
	L_1	0.055	0.117	0.073	0.007	0.077	0.005	0.289	0.003
	χ^2	**0.057**	0.124	0.064	0.009	0.094	0.006	0.264	0.003
t04d	L_2	0.018	0.051	0.023	0.029	0.000	0.007	0.050	0.001
	L_1	**0.021**	0.057	0.025	0.035	0.000	0.006	0.058	0.005
	χ^2	0.018	0.047	0.023	0.026	0.000	0.007	0.057	0.001
t03d	L_2	0.034	0.035	0.022	0.006	0.131	0.005	0.053	0.002
	L_1	**0.040**	0.037	0.026	0.006	0.154	0.009	0.055	0.002
	χ^2	0.028	0.032	0.019	0.006	0.092	0.009	0.055	0.002
t02s	L_2	**0.029**	0.057	0.034	0.129	0.012	0.008	0.000	0.015
	L_1	**0.029**	0.052	0.036	0.126	0.016	0.009	0.000	0.011
	χ^2	**0.029**	0.052	0.031	0.136	0.008	0.008	0.000	0.018

Table 3.8: Comparison of image distance metrics.

Table 3.9 compares several fusion functions that are used to merge the retrieval outputs from multiple query images. The distance metric is set to be the L_1 distance. It can be observed that the harmonic mean and maximum functions outperform the other fusion functions in terms of mean average precision. Their superior performance can be attributed to a nice property: they tend to give a higher rank to the images that are very close to one of the query images, even if they are far away from other query images. In some sense, the harmonic mean and maximum functions are similar to a noisy “logical-OR” operator on a set of boolean similarity predictions. This property is extremely important especially when relevant images only share similar visual patterns with *one* of the query images rather than *all* of them.

Data	Merge	MAP	P30	P100	Person	S-Obj	G-Obj	Sports	Others
t05s	Harmonic	0.032	0.090	0.057	0.007	0.021	0.003	0.194	0.004
	Maximum	**0.035**	0.110	0.064	0.019	0.019	0.003	0.194	0.004
	Minimum	0.002	0.017	0.015	0.004	0.000	0.001	0.000	0.000
	Average	0.027	0.079	0.049	0.002	0.009	0.003	0.190	0.003
	Product	0.031	0.085	0.054	0.003	0.024	0.003	0.192	0.004
t04s	Harmonic	0.020	0.048	0.037	0.007	0.000	0.001	0.129	0.002
	Maximum	**0.022**	0.055	0.039	0.004	0.000	0.001	0.152	0.004
	Minimum	0.001	0.003	0.004	0.000	0.000	0.001	0.000	0.002
	Average	0.013	0.033	0.027	0.006	0.000	0.001	0.083	0.001
	Product	0.017	0.042	0.030	0.007	0.000	0.001	0.110	0.002
t03s	Harmonic	**0.044**	0.087	0.057	0.000	0.100	0.008	0.247	0.009
	Maximum	0.037	0.071	0.052	0.000	0.102	0.006	0.158	0.007
	Minimum	0.002	0.005	0.003	0.000	0.000	0.005	0.000	0.002
	Average	0.034	0.076	0.053	0.000	0.057	0.003	0.252	0.009
	Product	0.038	0.080	0.057	0.000	0.073	0.004	0.251	0.009
t05d	Harmonic	0.058	0.125	0.075	0.007	0.072	0.007	0.309	0.004
	Maximum	**0.084**	0.169	0.084	0.034	0.125	0.025	0.328	0.016
	Minimum	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
	Average	0.049	0.121	0.064	0.003	0.051	0.005	0.290	0.003
	Product	0.053	0.111	0.066	0.003	0.063	0.005	0.299	0.003
t04d	Harmonic	0.022	0.058	0.027	0.033	0.000	0.007	0.066	0.005
	Maximum	**0.026**	0.065	0.029	0.039	0.000	0.008	0.074	0.005
	Minimum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Average	0.008	0.021	0.015	0.004	0.000	0.000	0.043	0.000
	Product	0.017	0.046	0.023	0.026	0.000	0.001	0.059	0.001
t03d	Harmonic	**0.040**	0.043	0.026	0.006	0.154	0.009	0.061	0.002
	Maximum	0.039	0.036	0.019	0.006	0.162	0.009	0.027	0.002
	Minimum	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Average	0.028	0.033	0.023	0.002	0.109	0.003	0.053	0.002
	Product	0.035	0.039	0.025	0.006	0.135	0.005	0.054	0.002
t02s	Harmonic	0.029	0.051	0.037	0.120	0.019	0.009	0.000	0.010
	Maximum	0.025	0.044	0.031	0.111	0.010	0.008	0.000	0.007
	Minimum	0.002	0.003	0.002	0.002	0.001	0.001	0.000	0.001
	Average	0.024	0.035	0.028	0.129	0.003	0.004	0.000	0.009
	Product	**0.030**	0.056	0.035	0.127	0.020	0.008	0.000	0.010

Table 3.9: Comparison of query example fusion strategies.

3.2.3 Combination with Text Retrieval

Combining image retrieval with complementary text retrieval results has proved to be an important and effective way to improve the performance in video retrieval. This problem was actively investigated by a large number of researchers in recent years. A detailed discussion on the combination approaches can be found in the previous chapter. In this section, we investigate two types of combination strategies (query independent and query-class dependent strategies) based on the weighted linear combination of text retrieval and image retrieval results. In more detail, let s be the overall retrieval score, s_t be the text retrieval score, s_i be the image retrieval scores and λ be the combination factor, then we decide the overall retrieval score by the following formula

$$s = s_t + \lambda s_i,$$

where λ varies from 0 to 1 in the experiments. McDonald et al. [MS05] showed that the weighted sum scheme is among the most effective approaches for text/image retrieval combination. Before the combination process, the confidence scores from different modalities/models usually need to be normalized into a uniform output space. Typical normalization schemes include rank normalization [YYH04], range normalization [AHI$^+$03] and logistic normalization [Pla99]. In this study, we choose rank normalization to calibrate the retrieval outputs. Figure 3.2 shows three types of learning curves (MAP, Prec@30 and Prec@100) with different combination factors based on query-independent combination strategy, namely, the combination is blindly applied for all queries on the TRECVID'03-'05 data collections. In lieu of being consistently improved with larger combination factors λ, the retrieval performance after combination usually drops when λ is larger than a small value around 0.1 - 0.2. For the collections of t04s and t04d, the degradation is pretty noticeable where the average precision at $\lambda = 1$ is even worse than using text retrieval alone. This indicates the query-independent combination strategy needs to be further refined in order to provide a consistent performance improvement. In contrast, Figure 3.3 shows the corresponding learning curves based on a simple query-class dependent combination strategy, namely, the combination is only applied on the queries that belong to the classes of finding specific objects and persons. As can be seen, this simple query-class combination strategy is effective and consistently achieves higher average precision with higher combination factor on image retrieval. The overall improvement is around a reasonable 2% given the relatively inferior performance of image retrieval techniques. These series of experiments again demonstrate the effectiveness of handling the combination method in a query-dependent way. Moreover, given the huge number of retrieval sources available in a multimedia retrieval system, query-class dependent combination will have greater potential to produce better retrieval outputs and this has been validated by many previous studies [YYH04, CNL$^+$04, KNC05].

3.3 Semantic Concept Detection

The image/video analysis community has long struggled to bridge semantic gap from successful, low-level feature analysis (color histograms, texture, shape) to

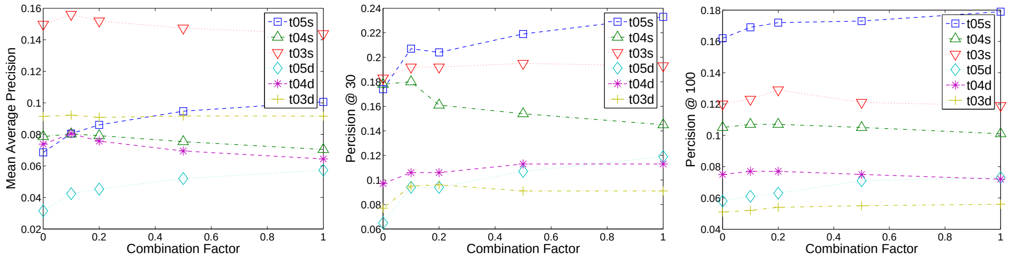


Figure 3.2: Comparison of combination factors λ (query independent combination).

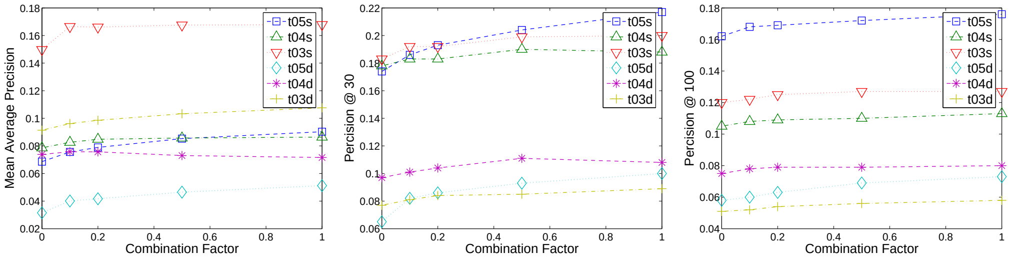


Figure 3.3: Comparison of combination factors λ (query dependent combination).

semantic content description of video. To overcome this gap, one approach is to utilize a set of intermediate semantic concepts [NS04b] that can be used to describe frequent visual content in video collections (e.g. outdoors, faces, animals). They can be seen as an intermediate step in enabling semantic video search and retrieval. These semantic concepts cover a wide range of topics [CMC05] such as those related to people (face, anchor, etc), acoustic (speech, music, significant pause), objects (image blobs, buildings, graphics), location (outdoors/indoors, cityscape, landscape, studio setting), genre (weather, financial, sports) and production (camera motion, blank frames). The task of automatic semantic concept detection has been investigated by many studies in recent years [BDF$^+$02, NKFH98, LTS03, YN05b, YCH04, JLM03, WCCS04, SP02, SSL02, VJZ98, VFJZ99]. Their successes have demonstrated that a large number of high-level semantic concepts are able to be inferred from the low-level multi-modal features of video collections.

Typically, the first step of developing a semantic concept detection system is to define a meaningful and manageable list of semantic concepts based on human prior knowledge. For each individual concept in the list, we should manually col-

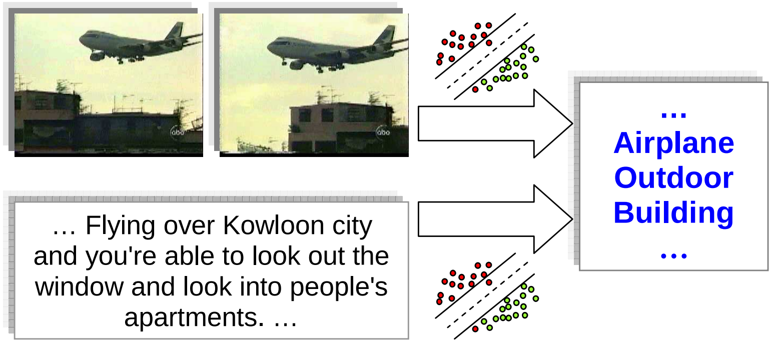

… Flying over Kowloon city and you're able to look out the window and look into people's apartments. …
…
Airplane
Outdoor
Building
…

Figure 3.4: Illustration of detecting semantic concepts from multimedia data. Each multimedia document is associated with multi-modal information such as text/speech transcript and visual frames. The semantic concepts can be detected by combining the outputs of multiple classifiers.

lect its ground truth annotations on a development video collection. For example, the common annotation forum in TRECVID'03 has successfully annotated 831 semantic concepts on a 65-hour development video collection [LTS03]. Some sample definitions of semantic concepts excerpted from TRECVID guidelines are listed as follows,

- **Waterscape**: segment contains video of a waterscape or waterfront
- **Mountain**: segment contains video of a mountain with slope(s) visible
- **Sports**: segment contains video of any sport in action
- **Car**: segment contains video of an automobile

Roughly speaking, we can categorize the semantic concepts into two types. One type consists of general concepts with frequent appearances in the video collection and therefore there are sufficient training examples to represent their characteristics. These concepts can often be learned with a reasonable prediction accuracy. For instance, in broadcast *news collection, anchor person, outdoors, cars and roads* belong to this type of concepts. In contrast, the other type of concepts consists of more specific concepts with less frequent occurrence. Thus, the number of their training examples is usually insufficient and less representative. In some sense, the detection of rare concepts is similar to a retrieval problem (with few training examples) rather than a classification problem. *Prisoner, physical* *violence* are two examples of this type of semantic concepts. The distinctions between these two concept types consequently suggest different roles in the retrieval task. For instance, the common semantic concepts often have universal predictive powers over a large number of queries, and their association with query topics can probably be learned from a large training collection. On the other hand, the usefulness of rare semantic concepts is limited to merely a small number of domains. Therefore, they are more appropriate to be applied in domain-specific queries. More details of the distinctions between common and rare semantic concepts can be found in [CH05].

In the rest of this section, we first provide an overview of the general (automatic) concept detection approaches including the steps of feature extraction, concept learning and multi-modal fusion. Then we discuss several open research directions related to the task of semantic concept detection.

3.3.1 General Approaches

Many previous methods approach the concept detection task as a supervised pattern recognition problem that attempts to discriminate the positive and negative annotated examples through automatically extracted low-level features. Typically, these learning algorithms intend to merge complementary prediction results from multiple modalities so as to boost the learning performance. Figure 3.4 illustrates the typical process of detecting semantic concepts from video sequences based on visual, audio and text/speech features. First, a variety of low-level features from each video clip are extracted from several modalities, e.g., text modality and visual modality. For each concept, separate uni-modal classifiers are built using the corresponding annotated data and low-level features. Then, a composite classifier is generated by combining multiple classifiers and it is applied to predict the presence/absence of semantic concepts on a large testing collection. Since detection results of semantic concepts are not related to any query topic, they can be indexed offline without consuming any on-the-fly computation resources. Such detection approaches have been applied in most extant multimedia semantic concept extraction systems [HBC$^+$03, AHI$^+$03].

A conventional semantic concept detection algorithm is made up of three major building blocks: low-level feature extraction, uni-modal feature-based learning and multi-modality fusion. Let us begin with discussing the first component, i.e., low-level feature extraction. So far, the most widely-used low-level features might include the following,

- Visual features: similar to content-based image retrieval, visual features can include the features based on color, texture, shape and so on. They can extracted from either the entire image, fixed-sized patches/blocks, segmented image blobs or automatically detected feature points;

- Text features: similar to text retrieval, text features can be extracted from ASR/CC transcript, video OCR and production metadata;

- Audio features: typical audio features are generated based on Short Time Fourier Transform(STFTs) including FFT, MFCC coefficients together with other features such as zero crossing, spectral centroid, pitch and etc.;

- Motion features: they can be represented in form of kinetic energy which measures the pixel variation within a shot, motion direction and magnitude histogram, optical flows and motion patterns in specific directions.

- Metadata: the metadata, also called surface features, are provided supplementarily in the production process, such as the name, the time stamp, the source of the video clips, the duration and location of video shots, and so forth. They are not sufficiently useful on their own but they can provide extra information to text/visual features.

Based on the aforementioned low-level features, the concept detector can be constructed using statistical learning algorithms. One of the most common learning algorithms is called support vector machines(SVMs) [Joa98, Vap95, LH02, CNL+04], which have been proposed with sound theoretical justifications to provide a good generalization performance compared to other algorithms [Bur98]. Built on the structural risk minimization principle, SVMs aim to seek a decision surface that can separate the data points into two classes with a maximal margin between them. The decision function is of the form,

$$y = sign\left(\sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b\right), \tag{3.8}$$

where x is the d-dimensional vector of a test example, $y \in \{-1, 1\}$ is a class label, x_i is the vector for the i^{th} training example, N is the number of training examples, $K(x, x_i)$ is a kernel function, $\alpha = \{\alpha_1, ..., \alpha_N\}$ and b are the parameters of the model. The kernel function can have different forms, such as the polynomial kernel $K(u, v) = (u \cdot v + 1)^p$ and the Radial Basis Function (RBF)

Concept	Avg Prec	Positive	Concept	Avg Prec	Positive
Person	0.8531	31161	Road	0.2481	2665
Face	0.7752	17337	Microphone	0.1947	2659
Outdoor	0.7114	15290	Interview	0.3019	2619
Studio	0.7541	4743	InterviewSeq	0.5237	2523
Building	0.3048	4177	Car	0.3151	2492
Female	0.2632	3887	Meeting	0.1708	2262
Walking	0.1635	3828	Anchor-Studio	0.8247	2392
Urban	0.1127	3586	Artficial-Text	0.6783	2373
Leader	0.1822	3033	Trees	0.2522	2152
Politicians	0.2782	2850	Sports	0.4481	1249
Asian-people	0.4247	2776	Maps	0.4816	610

Table 3.10: Average precision of using SVMs to detect 22 frequent semantic concepts. The model is learned from the TREC'05 development data with color moment features. The column "positive" indicates the number of positive examples of each concept out of 55932 training documents.

$K(u, v) = exp(-\gamma\|u - v\|^2)$ kernel[4]. Apart from SVMs, there are a large variety of classifiers that have been investigated in the context of semantic concept detection, including Gaussian mixture models(GMM) [AHI$^+$03, Wes04], hidden Markov models(HMM) [PGKK05], k Nearest Neighbor(kNN) [SBM05], logistic regression [HCC$^+$04], Adaboost [ZCZ$^+$04] and so on. To illustrate, Table 3.10 shows the average precision of detecting several frequent semantic concepts for the TREC'05 development data using SVMs [5].

To further refine the detection outputs, it is beneficial to combine the prediction outputs from multiple modalities that provide complementary information with each other. Generally speaking, there are two families of multi-modal fusion approaches, i.e., early fusion and late fusion. The early fusion method begins with merging the multi-modal features into a longer feature vector and takes it as the input of the learning algorithm. In contrast, the late fusion method directly fuses the detection outputs from multiple uni-modal classifiers. Both fusion methods have their own strengths and weaknesses [SWS05]. Early fusion combines all the information together and thus implicitly models the relations between different feature components. However, early fusion might face troubles if the feature

[4] Generally, the RBF kernel is most widely used due to its flexibility to capture the non-linear decision boundary and the good generalizability for the testing data. But be cautious that the parameter setting of the RBF model is critical to its classification performance.

[5] Thanks to Ming-yu Chen for providing the statistics of the semantic concept detection results.

constitution of different modalities is too heterogeneous with skewed length distribution and different numerical scales. But, this is less of a problem for late fusion, since the features from each modality will not interact with each other before the final fusion stage. Moreover, late fusion allows the system to adopt various detection techniques according to specific feature types. Also, it usually requires less computation power compared with the early fusion counterpart. Therefore, late fusion techniques appear to be more popular and are more extensively studied than early fusion techniques in the literature.

For late fusion approaches, the confidence scores generated from different modalities/models usually needs to be normalized before the fusion process. Typical normalization schemes include rank normalization [YYH04], range normalization, logistic normalization [Pla99], Gaussian normalization [AHI$^+$03]. The final detection results are then produced by merging the normalized confidences. In this step, it is reasonable to choose one of these choices: 1) combine multiple detection models, 2) combine the same detection models with different underlying features, or 3) combine the models with the same underlying features but different parameter configurations. Among all combination approaches, the simplest ones are based on manual definition. For instance, Amir et al. [AHI$^+$03] studied the min, max and unweighted linear combination function for multi-modality and multi-model fusion. However, in order to gain further improvement, a large body of machine learning methods have also been proposed and studied before. For instance, a more advanced multi-modal fusion strategy [WCCS04] called super-kernel fusion has been proposed, of which the underlying idea is to construct a hierarchy of kernel machines to model the complex decision boundaries beyond the linear combination. Yang et al. [YCH04] specifically considered the problem of detecting news subjects in news video archives by linearly combining the multi-modal information in videos, including transcripts, video structure and visual features. The weights are learned from SVMs. Cees et al. [SWS05] compared the early fusion and late fusion methods by using SVMs as the base classifiers and meta-level classifiers for fusing text and images. Their experiments on 184 hours broadcast video and 20 semantic concepts shows that late fusion on average tends to get slightly better performance than early fusion for most concepts, but if the early fusion is better for a concept, the improvement will be more significant than later fusion.

Due to the space limit, we refer to a survey written by Naphade [NS04b] for more details of the general concept detection strategies. Note that, in order to utilize additional domain knowledge for some widely applicable concepts such as faces, cars and sport events (soccer goal / basketball scores), researchers have also developed numerous domain-specific recognition approaches in a case-by-case basis. But since the discussions on various domain-specific techniques are outside the scope of this thesis, we will skip the discussions on this topic.

3.3.2 Balancing Rare Data

Theoretically, semantic concept detection can be handled by any supervised learning algorithm. However, this claim is not always valid in the real-world scenario, because learning algorithms often assume the positive/negative data distribution is balanced but multimedia collections usually contain only a small fraction of positive examples for semantic concepts. For example, there are only less than 8% shots labeled as cityscape and less than 3% labeled as landscape in the development set of the TRECVID'02. This is because the positive examples of a semantic concept is typically a coherent subset of images (e.g. cityscape, landscape and sunrise), but the negative class is less well-defined as "everything else" in the collection. Unfortunately, many learning algorithms will get into trouble when dealing with imbalanced datasets [Pro00]. For instance, when the class distribution is too skewed, SVMs will generate a trivial model by predicting everything to the majority class, even though SVMs have been shown to be relatively insensitive to the distribution of training examples. Japkowiczaz [Jap00] shows that the data imbalance issue can significantly degrade the prediction performance especially when training data are non-linearly separable. Therefore, it is of crucial importance for us to address the rare data problem in the context of detecting semantic concepts.

To date, there have been a few attempts to address the rare class problems in several real-world applications, such as fraud detection [CS98], network intrusion, text categorization and web mining [JAK02]. Two of the most popular solutions are named "over-sampling" which replicates the positive data, and "under-sampling" which throws away part of the negative data. They were designed to balance the data distribution and thus mitigate the data skewness problem in the training collection [WP01]. Although it is still an open question if artificially varying the training distribution can improve the prediction performance, Foster [WP01] provided insights and qualitative analysis of the effectiveness in tuning the training distribution. To demonstrate the effect of varying data distributions, we apply over-sampling to the TRECVID'02 data using SVMs, altering the positive data distribution from 10% - 60%. Figure 3.5 shows the detection performance for "cityscape" with respect to precision, recall and F1-measure. We observe that SVMs always predict test examples as negative and thus yields zero

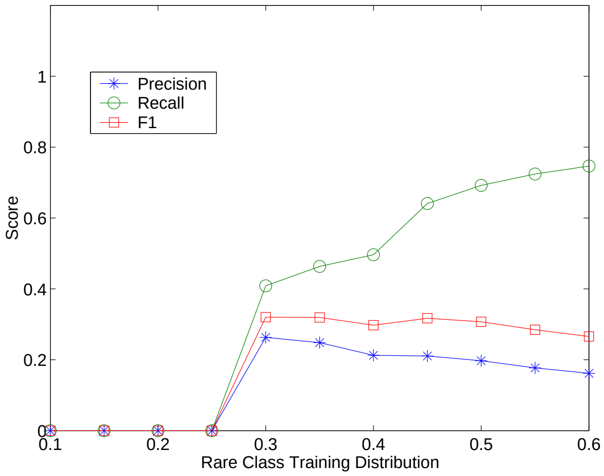


Figure 3.5: Effectiveness of modifying training distributions.

precision/recall until the size of the rare class examples is roughly comparable to the size of negative class examples. This observation again suggests that balancing the training class distribution is useful to improve the detection performance.

However, both under-sampling and over-sampling bear their known drawbacks. Under-sampling is likely to eliminate some of the potentially useful examples and such a loss of information might hurt the performance of classifiers. Over-sampling, on the other hand, significantly increases the number of training data and thus consumes more time in the learning process. This problem is more critical to SVMs than other learning algorithms, since the training time complexity for SVMs is close to quadratic of the number of support vectors [6], even cubic in the worse case [Joa95]. In addition, overfitting is more likely to occur with replicating minor examples [WP01].

As an alterative way to handle the skewed data distribution, ensemble-based approaches have been proposed in several recent studies, of which the basic idea is to combine multiple individual classifiers on balanced data distributions. In [CS98], a multi-classifier meta-learning approach has been devised to deal with skewed class distributions. Joshi et al. [JAK02] provided insights into the cases when AdaBoost, a strong ensemble-based learning algorithm, can achieve better precision and recall in the context of rare classes. They claimed that the performance of AdaBoost for rare class is critically dependent on the learning abilities of the base

[6] Actually a recent version of SVMs learning approaches (presented by Joachims in KDD'06) achieved a time complexity which is linear to the support vector number.

classifiers. To bring the strengths of under-sampling and over-sampling together, Yan et al. [YLJH03] proposed an ensemble approach that first partitions negative data into small groups, constructs multiple classifiers using positive data as well as each group of negative data, and finally merges them via a top-level meta-classifier. Various classifier combination strategies are investigated including majority voting, sum rule, neural network and hierarchical SVMs. Experimental results show that this approach can achieve higher and more stable performance than over/under-sampling strategies in the TRECVID datasets.

Beyond the approaches of augmenting learning algorithms, we can also consider other solutions from the perspective of leveraging multimodal information. For example, it is possible to perturb the original positive examples (by adding white noises or information from other modalities) and create a larger set of synthetic positive examples so as to balance the data distribution. In this scenario, how to produce the semantically correct but visually distinctive examples will become a major challenge.

3.3.3 Leveraging Unlabeled Data

Successful concept detection outputs usually rely on a large annotated training corpus that contains a sufficient number of (accurately labeled) image/video samples. Unfortunately in practice, the number of labeled video samples is usually not large enough for most semantic concepts, since manual annotation is such a labor-intensive process. For instance, annotating 1 hour of broadcast news video, with a lexicon of 100 semantic concepts can take anywhere between 8 to 15 hours [LTS03]. This problem is further worsened given a large number of infrequently-appearing semantic concepts in video collections.

As a remedy for the sparseness of labeled data, a variety of semi-supervised learning algorithms have been developed in an attempt to leverage additional unlabeled data in the training collection. Moreover, the multiple modalities in video streams further prompt us to consider the multi-view learning strategy which can explicitly split the feature space into multiple subsets, or views. Combining semi-supervised learning and multi-view setting offers more powerful tools to learn with unlabeled data and these approaches are generally called “multi-view semi-supervised learning”. Co-training [BM98] is one of the most well-known multi-view semi-supervised learning algorithms. It starts with two initial classifiers learned from a separate view. Both classifiers are then incrementally updated in every iteration using an augmented labeled set, which includes additional unlabeled samples with the highest classification confidence in each view.

Co-EM [NG00] can be viewed as a probabilistic version of co-training, which requires each classifier to provide class probability estimation for all unlabeled data. Collins and Singer [CS99] introduced the CoBoost algorithm which attempts to minimize the disagreement on the unlabeled data between classifiers of different views. This class of co-training type algorithms has been successfully applied to a variety of real-world domains, from natural language processing [PC01], web page classification [BM98], information extraction [CS99] to visual detection [LVF03].

However, these methods have not been successfully applied in the domain of video concept detection yet, although it has been considered a potential application domain by Blum et al. [BM98]. After examining the real-world video data, we realized that the failure of co-training in this domain can partially attributed to the violation of its underlying assumptions which requires that each view be sufficient for learning the target concepts. For example, when color histograms are used to learn the video concept "airplane", of two video frames that have the same color histogram, one might contain an airplane but the other might contain an eagle. Therefore, in this case, the view from low-level color features will not be sufficient to learn the underlying concept. Since most concepts describe the semantic contents instead of simple visual appearances, it is difficult for low-level visual features alone to sufficiently represent the concepts. In this case, co-training will usually exhibit poor performance because its assumptions are violated. In our previous work [YN05b], we found that co-training tends to produce lower average precision with more unlabeled data introduced with noisy labels. In the domain of natural language processing, Pierce et al. [PC01] also observed the similar degradation of the co-training algorithm if the labeled data introduced by the other view is not accurate enough.

Better semi-supervised learning algorithms should be able to guarantee that unlabeled data will at worst result in no significant performance degradation and at best improve performance over the use of the labeled data sets alone. Yan et al. [YN05b] proposed a more effective algorithm called semi-supervised cross feature learning(SCFL) for concept detection. Unlike co-training which updates each classifier by incorporating the selected unlabeled data to augment the labeled set, SCFL learns separate classifiers from selected unlabeled data and combines them with the classifiers learned from noise-free labeled data. One advantage of the proposed approach is that it can theoretically prevent its performance from being significantly degraded even when the assumption of view sufficiency fails. However, we also notice an overfitting effect for SCFL after more than a sufficient number of unlabeled data are added. Deciding the optimal stopping criteria can

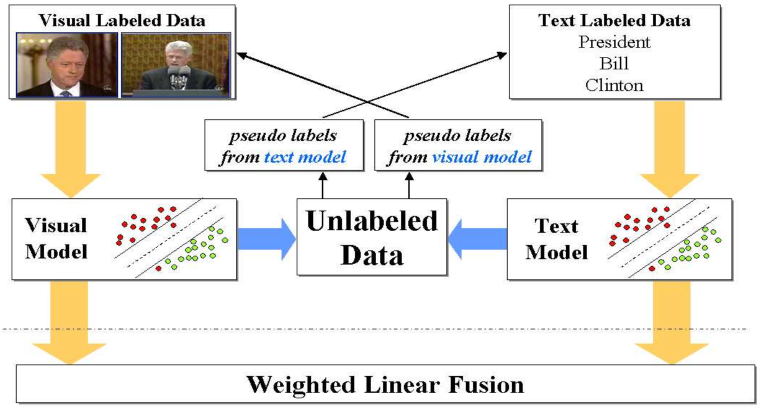


Figure 3.6: Illustration of co-training in multi-modal learning.

be based on validation set performance but it is left to future work.

If further manual annotation is possible, we can enhance the semi-supervised learning by iteratively inquiring a human annotator to review and provide the correct labels for some selected unlabeled data. This type of problem is called "active learning" [CCHW05] or "selective sampling" [CCS00] in the literature. An active learner begins with a pool of unlabeled data, selects a set of unlabeled examples to be manually labeled as positive or negative and learn from the newly obtained knowledge repetitively. Typically, the unlabeled examples can be selected by means of either minimization of the learner's expected error [CCS00] or maximization of information gain / version space reduction [TC01]. The effectiveness of active learning for reducing annotation cost in semantic concept detection has been demonstrated by previous work [CCHW05, NS04a, YH04, TC01]. By combining the idea of active learning and co-training, researchers proposed two different learning algorithms called corrected co-training [PC01] and co-testing [MMK02], which require users to annotate the selected unlabeled data from the co-training algorithm. They can be viewed as an empirical upper bound of the conventional co-training algorithm. Experimental results of applying corrected co-training [YN05a] to semantic concept detection is quite promising, which shows corrected co-training provides a considerable performance improvement over initial classification results.

3.3.4 Multi-Concept Relationship Modeling

Many concept detection approaches start by decoupling the set of semantic concepts and translating the learning task into multiple binary classification problems with the presence/absence label of each individual concept. However, these approaches simply ignore an important fact: *semantic concepts do not exist in isolation to each other*. They are interrelated and connected by their semantic interpretations and hence exhibit certain co-occurrence patterns in video collections. For example, the concept "car" always co-occurs in a video clip with the concept "road", while the concept "office" is not likely to appear with "bus". Such kind of concept relationship is not rare and it can be expected that mining multi-concept relationship can serve as a useful source of information to improve the concept detection accuracy. Moreover, such a correlated context could also be used to automatically construct a semantic network tailored to the video collection in a bottom-up manner. This semantic network construction could be helpful to discover unknown concept relationships that are complementary to human prior knowledge.

To automatically exploit benefits from this multi-concept relationship, several approaches have been proposed that are built upon advanced pattern recognition techniques. For example, Naphade et al. [NKFH98] explicitly modeled the linkages between various semantic concepts via a Bayesian network that implicitly offered ontology semantics underlying the video collection. Cees et al. [SWG$^+$04] proposed an semantic value chain architecture including a multi-concept learning layer called the context link. At the top level, it aims to merge the results of content outputs from concept detectors. Two configurations were explored where one was based on a stacked classifier upon a context vector and the other was based on ontology with some common sense rules. Hauptmann et al. [HCC$^+$04] tried to capture the inter-concept causations and fuse the multi-concept predictions by constructing an additional logistic regression classifier atop the uni-concept detection results. Amir et al. [AHI$^+$03] concatenated concept prediction scores into a long vector called model vectors and stacked a support vector machine on top to learn a binary classification for each concept. A ontology-based multi-classification algorithm was proposed by Wu et al. [WTS04] which attempted to model the possible influence relations between concepts based on a predefined ontology hierarchy.

In this section, we describe and compare some general multi-concept relational learning approaches via a unified probabilistic graphical model representation. Their effectiveness in video semantic concept detection is evaluated and

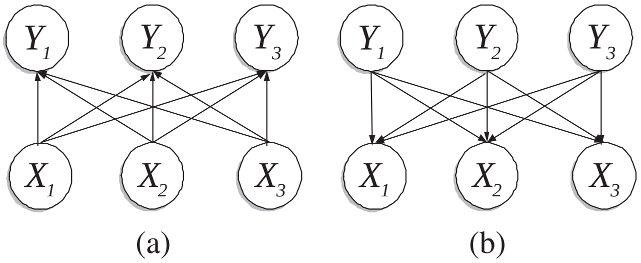


Figure 3.7: Two directed graphical models (a.k.a. Bayesian network) for multi-concept learning including (a) a generative model, and (b) a discriminative model, where X represents the observed detection outputs and Y represents the truth concept labels.

compared on two TRECVID'05 video collections.

Graphical Model Representations

Many multi-concept learning approaches can be concisely represented in the form of probabilistic graphical models that express dependencies among random variables by a graph in which each random variable is a node. There are two types of graphical models including directed graphical models (a.k.a. Bayesian network) which represent a factorization of the joint probability of all random variables and undirected graphical models in which graph separation encodes conditional independencies between variables. In this paper, we consider building graphical models between concepts and predictions generated from existing uni-modal semantic detectors rather than low-level video features, because we want to reduce the feature dimension and computational efforts in the learning process. Formally for a specific video shot, let $X_i \in \mathcal{R}$ denote the observations of i^{th} uni-modal semantic detector, $Y_j \in \{0, 1\}$ denote the presence/absence labels of j^{th} concept. $\mathbf{X}, \mathbf{Y}$ represent the vectors of $\{X_i\}$, $\{Y_j\}$. For the purpose of parameter estimation, we assume there are D training data with truth annotations $\{\mathbf{X}_d, \mathbf{Y}_d\}$. In this setting, the purpose of concept detection is to predict the hidden concept labels from visible observations provided by uni-modal classifiers, i.e., estimate the conditional probabilities of $P(Y_j|\mathbf{X})$ under a given model representation. In the following discussions, we discuss some existing models and propose several new models for mining the multi-concept relationship using both the directed graphical models and the undirected graphical models.

Most detection approaches belong to the category of directed graphical models. Essentially, all these models can be understood as a two-layer directed graphical model with one layer of hidden units and one layer of input units connected by fully-linked edges. Figure 3.7 shows several examples of directed graphical models for video concept mining. Among them, Figure 3.7(a) corresponds to a generative model(**BNG**) that assumes the detection outputs are generated by concept variables. One such example is the Bayesian network version of the multi-net model proposed by Naphade [NKFH98]. In this model, the hidden layer can be taken as a representation of the "latent concept aspects" and the input layer corresponds to the observed predictions of uni-modal semantic detectors. This graph naturally implies the conditional independence of predictions X_i given the concepts $\mathbf{Y}$. Typically, the prior distribution of Y_j is modeled as a Bernoulli distribution $Bernoulli(p_j)$ with a draw probability p_j and the conditional probability of $X_i|Y_j$ is modeled as a Gaussian distribution with mean $\sum_j w_{ij}y_j$ and variance σ_i^2. The model parameters can be learned based on the maximum likelihood estimation(MLE),

$$\begin{aligned}(W, \Sigma, P) &= \arg\max \sum_d \log P(x_{d1}, ..., x_{dM}, y_{d1}, ..., y_{dN}) \\ &= \arg\max \sum_d \sum_i \log P(x_{di}|\mathbf{Y}) + \sum_d \sum_j \log P(y_{dj})\end{aligned}$$

By setting the derivatives of likelihood function with respect to each parameter to be zero, we can derive the maximum likelihood estimators in an analytical form where the estimated parameters are shown as follows,

$$\begin{aligned}p_j^* &= \frac{\sum_d I(y_{dj} = 1)}{D} \\ w_{ij}^* &= \arg\max_w \sum_d (x_i - \sum_j w_{ij}y_j)^2, j = 1...N \\ \sigma_i &= std(x_i)\end{aligned}$$

Note that the estimation of parameter w_{ij} is equivalent to a linear regression on X_i with underlying variables $Y_1, ..., Y_N$. Although the parameter estimation process is quite simple, it is usually intractable to infer the conditional probability of $\mathbf{Y}$ given $\mathbf{X}$ due to the lack of conditional independence between $\mathbf{Y}$. Therefore, we adopt a popular approximate inference technique called Gibbs sampling, which is applicable when the joint distribution is not known explicitly but the conditional distribution of each variable can be computed.[7] According to Gibbs sampling, we

[7] The Gibbs sampling algorithm is to generate an instance from the distribution of each variable

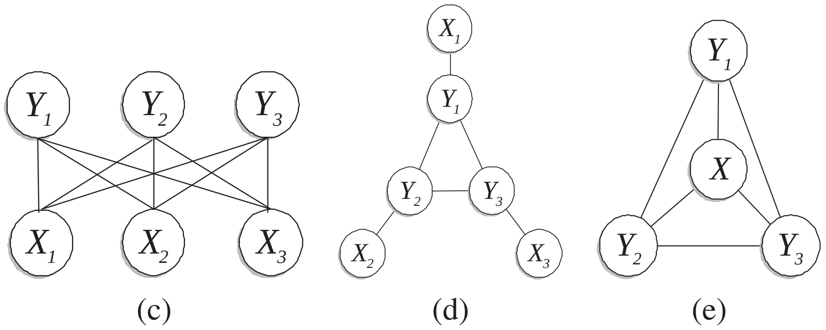


Figure 3.8: Three undirected graphical models for multi-concept learning including (a) restricted Boltzmann machines, (b) Markov random fields and (c) conditional random fields.

can repeatedly sample the following conditional probability to approximate the joint distribution and then compute the expectation of labels $\mathbf{Y}$,

$$Y_j \sim P(Y_j = 1 | \mathbf{X}, \mathbf{Y} \setminus Y_j) = \frac{P(Y_j = 1, \mathbf{X}, \mathbf{Y})}{P(Y_j = 0, \mathbf{X}, \mathbf{Y}) + P(Y_j = 1, \mathbf{X}, \mathbf{Y})}$$

The model in Figure 3.9(b) corresponds to another type of directed graphical models which directly model the conditional probability of $\mathbf{Y}$ given $\mathbf{X}$, also called discriminative models(**BND**). It can used to describe the approaches proposed in [HCC$^+$04, SWG$^+$04]. Unlike the previous models, this graph reversely implies the conditional independence of predictions Y_i given the observations $\mathbf{X}$ and thus results in an fast inference process. In practice, the labels Y are usually modeled by a logistic regression based on observation variables X where,

$$P(\mathbf{Y}|\mathbf{X}) \propto \exp\left[\sum_i (\alpha_i + \sum_j w_{ij} x_j) y_i\right]$$

With an additional validation set, the parameters can be estimated by using any gradient descent methods such as the iterative reweighted least squares(IRLS) algorithm [JJ94].

The Bayesian network formalism offers clear causal semantics and manipulability from a modeling point of view. However, as pointed out by [XYH05],

in turn, conditional on the current values of the other variables. It can be shown that the sequence of samples comprises a Markov chain, and the stationary distribution of that Markov chain is just the sought-after joint distribution.

inference of the latent concepts in such models can be prohibitively expensive due to the conditional dependencies between all hidden variables. This drawback could seriously affect the model performance in real-time prediction tasks and in EM-based learning. Moreover, directed models have to explicitly retain the causality between different observed/hidden variables and this can lead to a sophisticated network structure if we want to incorporate additional dependency between concepts.

As alternatives of directed graphical models, undirected graphical models could be a better formalism for handling the relation between concepts without explicitly imposing the concept causality. However to our surprise, the options of applying undirected models to video annotation have seldom been explored before. One example of undirected models is shown in the Figure 3.8(c) called the restricted Boltzmann machine(**RBM**)(a.k.a. harmoniums) which can be viewed as an undirected counterpart of the aforementioned directed concept models with the arrows of the edges removed. In a RBM model, observations X are fully connected with the concept presence Y in form of a bipartite graph. According to the undirected model semantics, there is no *marginal* independence for either input or hidden variables. However, it enjoys the advantages of *conditional* independence between hidden variables given observed variables, which is generally violated in the directed models. This property can greatly reduce inference cost although it comes at a price of a more difficult learning process due to the presence of a global partition function. Formally, we can define the conditional probabilities as follows,

$$P(x_i|\mathbf{Y}) = \mathcal{N}\left[\sigma_i^2(\beta_i + \sum_j w_{ij}y_j), \sigma_i^2\right], P(y_j|\mathbf{X}) = \mathcal{S}\left[\alpha_j + \sum_i w_{ij}x_i\right]$$

where $N(\mu, \sigma)$ is a normal pdf function with mean μ and variance σ, and $\mathcal{S}(x)$ is a logistic function $1/(1 + e^{-x})$. Then we can reconstruct the the joint probability of $\mathbf{X}, \mathbf{Y}$ to be,

$$P(\mathbf{X}, \mathbf{Y}) \propto \exp\left[-\frac{1}{2}\sum_i \frac{x_i^2}{\sigma_i^2} + \sum_i \beta_i x_i + \sum_j (\alpha_j + \sum_i w_{ij}x_i)y_j\right].$$

The gradient-descent learning rules can be obtained by taking derivatives of the log-likelihood with respect to the model parameters. It can be found that the gradient of each parameter are equivalent to the difference between expectation of its corresponding potential under empirical distribution and that under model distribution [XYH05]. However, the expectations under the model distribution are usually difficult to compute because of the intractable normalization factor. Therefore, we have to utilize some approximate inference approaches such as loopy belief propagation, contrastive divergence(CD) and variational methods. In practice, we adopt the contrastive divergence as the basic inference method [XYH05] which approximates the intractable model distribution using a single or a few iterations of Gibbs sampling, and is therefore highly efficient.

Until now, we have only discussed two-layer bipartite graphical models with the nodes in each layer are fully connected with the nodes in the other layer. Rather than modeling the concept relationship in such an indirect way, the flexibility of undirected models allow us to impose the links directly on the concept nodes which cannot be easily achieved by a directed model. We considered two possibilities of such kinds of model designs in the following discussions. Figure 3.8(d) corresponds to a Ising-model like Markov random field(**MRF**) where the concept nodes are fully linked and the observations only interact with their corresponding concepts. Based on the model semantics and similar potential definitions as before, the joint probability of the observations and labels can be represented as follows,

$$P(\mathbf{X},\mathbf{Y}) \propto \exp\left[-\frac{1}{2}\sum_i \frac{x_i^2}{\sigma_i^2} + \sum_i \beta_i x_i + \sum_i (\alpha_i + w_i x_i + \sum_j u_{ij} y_j) y_i\right].$$

The major difference between above equation and RBM lies in the pairwise interaction terms of $u_{ij}y_jy_i$ which directly captures the concept co-occurrence patterns. The maximum likelihood estimation of this model can also be achieved by contrastive divergence. Note that an advanced version of the multi-net model based on the factor graph model [NKFH98] can be viewed as a variant of above model with slight differences in the model presentation. Figure 3.8(e) plots a more recently developed graphical model called the conditional random field(**CRF**) which is a random field globally conditioned on the observations X. It means the observations Y, when conditioned on X, obeys the Markov property with respect to the undirected graph in Figure(e). According to the Hammersley-Clifford theorem and assuming only the pairwise clique potentials are nonzero, we can define the joint probability as,

$$P(\mathbf{Y}|\mathbf{X}) \propto \exp\left[\sum_i (\alpha_i + \sum_j w_{ij} x_j) y_i + \sum_i \sum_k u_{ik} y_i y_k\right]$$

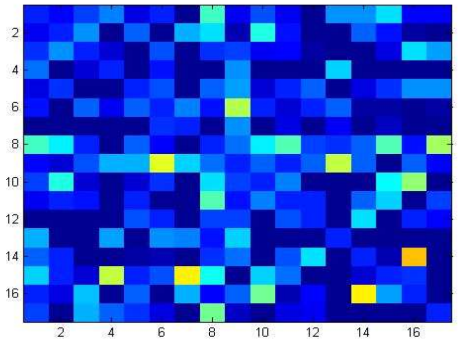


Figure 3.9: Illustration of inter-concept relationships learned from CRF. Each grid indicates a pair of concepts. Lighter colors stand for stronger positive relations and darker colors stand for stronger negative relations. For instance, "face" (concept 6) and "person" (concept 9) have a strong positive relation.

Method	**5-concept collection**			**11-concept collection**		
	Better	Worse	MAP	Better	Worse	MAP
Base	-	-	0.5715	-	-	0.4994
BNG	2	3	0.5742	5	6	0.5187
BND	4	0	0.6036	6	4	0.4996
RBM	4	0	0.6024	5	5	0.4822
MRF	3	1	0.5714	6	4	0.5210
CRF	3	1	0.5882	7	3	0.5211

Table 3.11: Comparison of multi-concept relational learning models. "Better/worse" means how many concepts have a better/worse performance than baseline. "MAP" means the mean average precision of each learning method.

The conditional random field takes the advantage of modeling the conditional probability of concepts given observations and thus avoid the problem of learning complex class density. Due to the space limit, please refer to [LMP01] for the details of learning and inference in the conditional random fields.

Experimental Results

We evaluated all five multi-concept detection models using the TRECVID'05 development data. The development data are split into three parts, where 70% as the training set to generate the concept detection outputs, 15% as the validation set to learn the multi-concept relationship, and remaining 15% as the testing set to evaluate the detection performance. As mentioned before, one application of multi-concept learning models is to automatically discover the co-occurrence patterns in a specific video collection. To illustrate this, we plotted the Figure 3.9 which shows the relationships between 17 concepts found by CRF with each grid indicating a pair of concepts. As can be seen, there are a considerable amount of strong correlations between semantic concepts in the video collection, including both positive interactions (two concepts are positively correlated with each other) and negative interactions (two concepts are negatively correlated). We believe the concept detection task should be able to benefit from capturing this semantic context. In more detail, some of the strongest positive/negative concept pairs are listed below,

Positive Pairs: (outdoor, building), (urban, building), (person, face), (studio, maps), (car, road), (urban, road), (text, sports)

Negative Pairs: (sports, building), (outdoor, computer tv screen), (outdoor, maps), (commercial, studio), (waterfront, urban)

However, we also notice that not every concept can exhibit co-occurrence patterns with other concepts due to the limited number of training data. It would be beneficial to remove those isolated concepts in the training data before the learning process. By conducting the χ^2 test between every pair of concepts, we eliminated the concepts that do not have any χ^2 scores exceeding certain thresholds and thus not strongly correlated to others. Finally, we constructed a five-concept collection and an eleven-concept collection that include the sets of concepts as follows,

5-concept: car, face, person, text, walking/running

11-concept: building, car, face, maps, outdoor, person, sports, studio, text, urban, walking/running

Table 3.11 shows the mean average precision on the testing set of the five graphical models discussed before and the baseline obtained without taking any conceptual relations into account. We can observe that the best multi-concept modeling approaches can usually bring an additional 2-3% improvement over the baseline performance in terms of mean average precision. Table 3.11 also lists the number of concepts that have better and worse performances than baseline. It can be found that the detection accuracy of each concept is more likely to be improved with aid of multi-concept relational modeling, which shows the effectiveness of incorporating contexts into the detection results. The undirected graphical models (i.e., RBM, MRF and CRF) demonstrate their promising potentials in the task of concept detection given the high MAP on both datasets. But on the other hand, our experimental results also show the inconsistency of the performance of various models. For example, in the 5-concept dataset, BND and RBM are among the best models with similar MAPs around 60%. But in contrast, in the 11-concept dataset MRF and CRF provide the best performance around 52%. After an in-depth analysis on this dataset, we found that the inferior performances of BND and RBM mainly come from some noticeable degradations on one or two concepts even they can improve on the others, which might indicate their instabilities when handling a large number of concepts. However, it is worth pointing out that so far the differences between models and baseline are not statistically significant yet. Further evaluations are suggested to provide more insights on their comparison.

3.3.5 Hidden Semantic Concept Extraction

Pre-defining a list of semantic concepts and learning from manually annotated data is the standard way to produce a large number of semantic concepts. However, this type of approach usually undergoes a process that requires labor-intensive manual annotations. Moreover, it is not straightforward to define a list of concepts which can both be detected accurately from low features and help users express the semantic meanings of most general queries. Therefore, it is instructive to discuss an alternative solution in an attempt to extract the hidden semantic concepts of multimedia documents in an unsupervised manner.

A starting point is to follow the idea of latent semantic indexing(LSI) [DDF$^+$90] in text retrieval. LSI begins with considering a feature matrix $M^{m \times n}$ where each row of the matrix M_i contains all low-level features of the i^{th} video documents in the video collection, m is the number of documents and n is the number of low-level features. To extract k hidden concepts from the matrix, LSI computes the Singular Value Decomposition(SVD) for the matrix M and keep the k largest singular values as well as the corresponding singular vectors,

$$M = USV^T \approx M_k = U_k S_k V_k^T$$

where S_k is a $k \times k$ diagonal matrix with k largest singular values, U_k is a $m \times k$ matrix with orthogonal columns corresponding to the mapping between documents and concepts, V_k is a $k \times n$ matrix with orthogonal columns corresponding to the mapping between concepts and features. The lower-rank matrix M_k is actually the closest rank-k approximation of the feature matrix M in terms of minimizing the mean square error between M and M_k. Each row of the left singular vector matrix U_k can be viewed as a vector of hidden semantic concepts in each video document. After the computation of SVD, the hidden semantic concepts are ready to be plugged into the probabilistic retrieval model, which serve as an additional set of video descriptors for the following learning algorithm.

Along this direction, a number of unsupervised learning approaches have been developed to model data with multiple modalities [XKC$^+$04, XYH05, BJ03] and capture the latent topics from unannotated multimedia archives. For example, Xie et al. [XKC$^+$04] presented a hierarchical HMM model(HHMM) for unsupervised mining of video temporal structures. The HHMM model can uncover the multi-level statistical structures based on a multi-level Markov assumption. Hofmann's probabilistic latent semantic indexing (pLSI) model [Hof99] extends the LSI idea to a probabilistic framework by assuming words to be marginally *iid* samples from a document-specific mixture of word distributions. The joint probability of a word w and a document d in pLSI model can be represented as,

$$p(d, w) = p(d) \sum_z p(w|z)p(z|d),$$

where z is the unobserved topic variable. The mixture of unigrams model [BNJ03] is a special case of pLSI where each document is associated with only one topic. The latent Dirichlet allocation (LDA) model by Blei et al. [BNJ03] offers a more expressive and generalizable text modeling scheme by associating with each document a unique latent *topic-mixing* vector represented by a random point (under some "prior" distribution) in a simplex. Each word is independently sampled according to different topic draw from the topic mixture, hence generatively, topic mixing is achieved at the document level when these words (possibly from different topics) are pooled. The joint probability of a topic mixture θ and a set of N word $\mathbf{w}$ in LDA model can be represented as,

$$p(\theta, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_n \sum_z p(z_n|\theta)p(w_n|z_n, \beta),$$

where z_n is a set of latent topics associated with each word w_n. It is straightforward to extend these models to handle multimedia data. Gemert [vG03] applied LSI to capture the joint latent semantic space of text and images. Blei and Jordan [BJ03] have extended the mixture of unigrams and the LDA model into a Gaussian-multinomial mixture (GM-Mix) and a Gaussian-multinomial LDA (GM-LDA) to model captioned images.

Essentially, all the aforementioned models can be understood as a two-layer directed graphical model (i.e., a Bayesian network) with one layer of hidden units and one layer of input units connected by edges pointing from the hidden layer. The hidden layer can be taken as a representation of the "latent topic aspects" and the input layer corresponds to the observed features of a sample such as a document. Such a Bayesian network formalism offers clear causal semantics and manipulability from a modeling point of view. However, as pointed out by Welling et al. [WRZH04], inference of the latent topics in such models can be prohibitively expensive due to the conditional dependencies between all hidden variables. This drawback could seriously affect the model performance in real-time prediction tasks and in EM-based learning. Therefore, Xing et al. [XYH05] presented a *dual-wing harmonium* (DWH) model for multimedia data based on a two-layer undirected graphical model called harmonium [Smo86, WRZH04]. This model can be viewed as an undirected counterpart of the aforementioned directed aspect models such as LDA, with the distinctions in topic representation and mixing scheme. This model enjoys some unique advantages: 1) inference is fast due to the conditional independence of the hidden units although the offline learning process could be difficult; 2) topic mixing can be achieved by document-specific and feature-specific combination of aspects rather than via a cumulative effect of single topic draws. But it might suffer from a longer learning process due to the existence of an intractable normalization factor.

To illustrate the possibility of automatically discovering meaningful latent topics from both text and images, we select 5 latent topics out of the 20 topics learned by DWH and show their associated terms/images in Fig 3.10. Each topic is described by the top 10 words and the top 5 key images associated with the video shots that provide the highest conditional probabilities on the latent topic. The examples of the first three topics clearly correspond to the scenes of Weather News, Basketball and Airplane, respectively, which is clustered based on the evidence from both words and images. The fourth topic is kind of narrow, which captures the scene of the same anchorperson from CNN Headline News. This topic is likely to be generated from the evidence primarily from image similarities. The last topic illustrates some interesting patterns. At a first sight, these shots appear to follow a degenerated theme because they cover different scenes including both basketball and weather news. But after reviewing the associated closed-captions, we found that these shots do share some common aspects. First, they all mention similar place names and numbers in the transcript such as "(New) York", "(New) Jersey" and "sixth". Second, both the weather news and basketball reports uses the same terms in reporting, such as "losing". Apparently, the learning algorithm

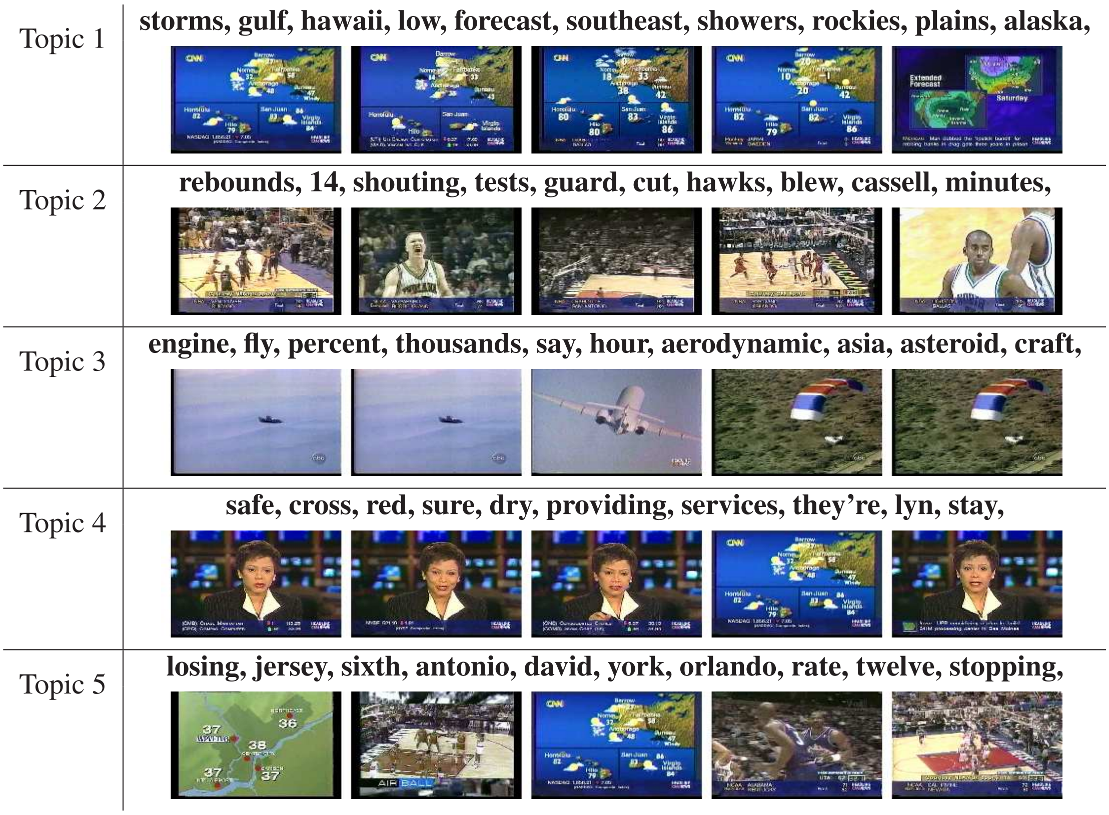

Topic 1
storms, gulf, hawaii, low, forecast, southeast, showers, rockies, plains, alaska,
Topic 2
rebounds, 14, shouting, tests, guard, cut, hawks, blew, cassell, minutes,
Topic 3
engine, fly, percent, thousands, say, hour, aerodynamic, asia, asteroid, craft,
Topic 4
safe, cross, red, sure, dry, providing, services, they're, lyn, stay,
Topic 5
losing, jersey, sixth, antonio, david, york, orlando, rate, twelve, stopping,

Figure 3.10: Illustration of five latent topics learned from video collections. Each topic is associated with the top 10 words and the top 5 images extracted from the most related documents.

discovers the last topic mainly based on the word similarities.

In summary, automatically extracting hidden concepts enjoys the advantage that no human annotations are required to be collected from the video collection. Also, it can discover some distinguishable concepts/clusters based on the separability on low-level features, instead of manually making up a list of semantic descriptors which could be difficult to learn on the low-level feature space. But one of its disadvantages is the lack of interpretability on the extracted hidden semantic concepts, because no explicit semantic meaning is associated with them before the learning process, although this weakness does not cause too much troubles for the learning algorithms. Therefore, in my opinion, if the precision of pre-selected semantic concepts is not sufficiently high to support the retrieval task, resorting to the automatically extracted hidden concepts turns out to be a more feasible choice. But this claim might still need additional empirical evidence to validate.

3.3.6 Associating Words with Images - Yet Another Way for Concept Detection

Viewing semantic concepts as binary detection outputs on low-level multi-modal features is not the only way to model the concept detection problem. Recently, an emerging direction for (image/video) concept detection is to jointly model the associations between annotated concept words and image features. These approaches typically assume the concept words and image features are generated by the same set of hidden information sources. Hence, image features and concepts are no longer marginally independent to each other. Once image features are given, associated concept words can be inferred by the information flow passed through the hidden layer. Actually, most of these approaches have been designed under a slightly different name called “image annotation”, which aims to match the associating text keywords to their corresponding image regions and automatically predict new keywords from given image regions. But if we consider semantic concepts as text keywords related to the image, the image annotation techniques can straightforwardly be reused.

A number of learning algorithms have been applied in the task of automatic image annotation, such as machine translation models [BDF$^+$02], relevance language models [JLM03], graphical models [BJ03] and graph random-walking methods [PYFD04]. Barnard et al. [BDF$^+$02] interpreted image annotation as a machine translation problem and solved it via two IBM translation models by learning their joint probabilities (with and without word orderings). Blei et al. [BJ03] developed a gaussian-multinomial latent Dirichlet allocation(GM-LDA) model and a correspondent latent Dirichlet allocation(GM-LDA) model that simultaneously capture the information from image regions and associated text keywords via a directed graphical model. Jeon et al. [JLM03] used the framework of cross-lingual retrieval to formulate the image/video annotation. They proposed an annotation model called cross-media relevance model(CMRM) which directly computed the probability of annotations given the image. It was shown to outperform the translation models in the image/video annotation task. By representing the terms and image features in a unified graph, Pan et al. [PYFD04] proposed a random walk with restart(RWR) approach to capture the correlation between words and images. Jin et al. [JCL04] proposed a coherent langauge model for image annotation that can model the word-to-word relationship in the annotation process. This approach allows the annotation length to be automatically determined and the annotated number of examples to be reduced by using active learning technique. Iyengar et al. [IDF$^+$05] described a joint text/image modeling approach for video retrieval that allows the full interaction between multi-modalities to result in a considerable performance improvement in TRECVID datasets.

3.3.7 Are Semantic Concepts Useful to Video Retrieval? - A Case Study

To illustrate the usefulness of constructing a large number of high-level semantic concepts to enhance video retrieval quality, we provide a case study based on a large-scale TRECVID video collection [SO03] and two recently developed video concept lexica [NST$^+$06, SWvG$^+$06], namely the Large-Scale Concept Ontology for Multimedia (LSCOM) and the MediaMill challenge concept data, both of which also include an “annotation corpus” for the TRECVID 2005 video collection, where for each concept in the lexicon and every shot in the collection, it was manually determined whether the concept was absent or present in the shot. Our video retrieval experiments are conducted on 83 use cases queries, developed as part of LSCOM (also with concept truth annotations available on TRECVID 2005 video) to uncover the relationship between the number of concepts, detection accuracy and retrieval performance. We also speculate what minimal number of concepts will be sufficient to construct a highly accurate video retrieval system.

Retrieval Experiments

To evaluate the retrieval utility of the LSCOM-Lite, MediaMill and the full LSCOM concept sets, we designed experiments based on the guidelines of the automatic retrieval task in TREC video retrieval evaluation (TRECVID), which requires an automatic video retrieval system to search relevant documents without any human feedback. As the baseline, we generated the standard text retrieval output by searching the speech transcript for automatically extracted text keywords from each use case. Given the annotations of high-level concepts and relevance judgments of use case queries at our disposal, we can linearly combine the semantic concept predictions into the text retrieval outputs so as to determine how much these use cases would be affected. In more detail, we compared text retrieval results to the results after incorporating (a) the LSCOM-Lite set of 39 concepts, (b) 75 concepts from the 101 concept MediaMill challenge which overlapped with the LSCOM concepts, and (c) 300 of the LSCOM concepts that had at least 10 positive instances in the annotations.

Let us begin our discussions with the most ideal case, where we assume that the semantic concept detection is perfect (equivalent to directly using the concept truth annotation) and that the combination method of concepts is optimal. Although impractical, the results reported in this setting can serve as a theoretical upper bound to indicate how useful the concepts can offer at most. In our previous work [YH03], we developed a theoretical framework for studying the performance limits over both monotonic and linear combination functions. In brief, the approach computes the optimal linear combination performance with the semantic concepts f_i fixed. Let us denote the linear combination $F(D,Q) = \sum_{i=1}^{N} \lambda_i f_i(D,Q)$ and $AP(F(D,Q))$ as the average precision of order list σ where σ is determined by retrieval score $F(D,Q)$ with respect to D. Therefore our task can be rewritten as a bounded constrained global optimization problem, $\max_{\lambda_i} AP\left(\sum_{i=1}^{k} \lambda_i f_i(D,Q)\right)$. To handle the bounded constraint optimization problem, we use a global optimization algorithm called the MCS algorithm, proposed by Huyer et al. [HN99]. Their method combines both global search and local search into one unified framework via multi-level coordinate search. It is guaranteed to converge if the function is continuous. Also in our implementation, multiple start points are tried for each query to avoid local minima problems.

To get a more realistic estimate (as opposed to the perfect "oracle" detection) of the concept utility with state-of-the-art concept detection approaches, we repeated the experiment after introducing noise into the perfect concept prediction (but still with a oracle combination). The results from TRECVID 2006 semantic concept classification evaluation show that the current best semantic concept classification systems can average slightly less than 0.2 MAP over the LSCOM-Lite concepts. Because mean average precision is a ranked-based measure and difficult to simulate, we approximated this MAP with a breakeven precision-recall point at 0.2[8]. This was easily achieved by randomly switching the labels of positively annotated shots to be (incorrectly) labeled as negative and conversely switching some negatively labeled shots as incorrect positive examples, until we achieve the desired breakeven point where precision is equal to recall. This made the concept labels appear roughly equivalent to a detector with MAP of 20%.

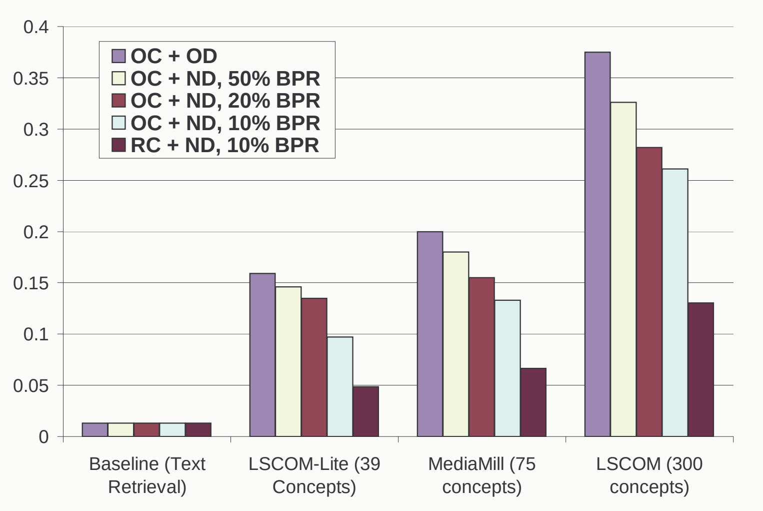


Figure 3.11: The oracle detection and oracle combination (OC + OD) results on the use cases for text-only retrieval and retrieval with the LSCOM-Lite, MediaMill and the larger LSCOM concept sets. The results using simulated "noisily detected" (OC + ND) concepts are also shown at 50%, 20% and 10% breakeven precision recall (shown as "50% BPR", "20% BPR" and "10% BPR" respectively. Realistic combination estimates (RC) with 10% breakeven precision recall are shown as "RC + ND, 10% BPR".

Figure 3.11 shows the retrieval results of combining semantic concepts for the three sets under different noise levels, where we can observe a substantial improvement brought by additional semantic concepts. For example, the video

[8] Breakeven precision-recall is usually a good approximation for mean average precision. They are equivalent to each other if the precision-recall curve is mirror symmetric to the line of precision = recall.

retrieval performance can reach an impressive 37% in terms of MAP after the full LSCOM set of concepts are incorporated, given that the baseline text-only retrieval MAP is only 1%. This surprisingly low baseline of the text-only retrieval, achieved using a configuration typical for video retrieval, can be attributed to the strong visual specificity of the queries, which cannot be answered by merely searching the audio transcript. This series of experiments confirms the huge potential of high-level concepts in helping to build an effective video retrieval system. As a more realistic setting, we also investigate the performance after introducing detection noise into the semantic concepts. In this case, the retrieval performance keeps decreasing when more and more positive shots are switched to negative ones. This shows that detection accuracy of semantic concepts has a noticeable effect on the quality of video retrieval accuracy. However, even if the breakeven precision-recall of these sets of concepts is only 10%, the retrieval MAP can still be boosted to 26% with the full set of LSCOM concepts. This suggests that although the prediction provided by the state-of-the-art automatic concept detection algorithms is far from perfect, they can still retain their potential in augmenting the standard text retrieval output.

It is worth mentioning that all of the above discussions assume the true relevant documents for each query are available and therefore we can use learning approaches to estimate the optimal combination model. However in practice, we will not be able to collect ground truth for every possible query that users may submit. Hence, it is more reasonable for us to derive the combination models based on some obtainable query properties, such as query description, query context, user profile, or interactive relevance feedback. Based on recent results reported in this thesis, realistic combination models (which are determined by the query description alone) may result in a 30%-50% loss of accuracy compared with the optimal combination models. Thus we modeled the "realistic combination" assumption in Figure 3.11 (and later in Figure 3.12) using a 50% degradation over oracle combination. However, even taking this additional discounting factor into consideration, the high-level concepts are still shown to be a potentially useful component for video retrieval given that it can boost MAP from 1% to above 10%.

How many concepts are sufficient for a good retrieval system?

One of the most interesting aspects of this work is that these results can now provide hints to answer the question: how many concepts are sufficient to construct a "good" video retrieval system? To be more rigorous in the rest of the discussions,

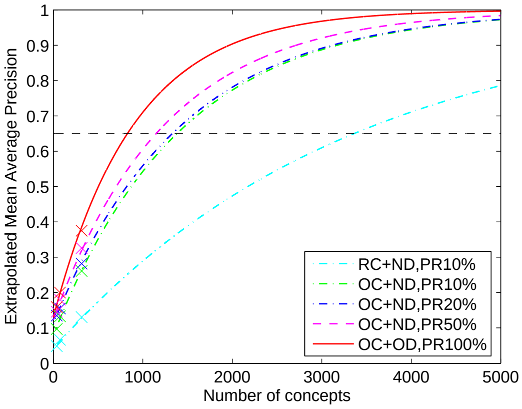


Figure 3.12: Extrapolated mean average precision vs. the number of high-level concepts. The true MAPs on three concept sets are indicated by "x". The corresponding fitted curves are also plotted, where the solid line means using the oracle detectors, and the dash line means using the noisy detectors with 50%, 20%, 10% breakeven precision-recall. Realistic combination estimates (RC) with 10% breakeven precision recall are also shown.

we define a "good" system as a system that can achieve more than 65% MAP. This corresponds to the current MAP accuracy that can be achieved by the best web (text) search engines [BJF$^+$05]. In order to investigate this problem more thoroughly, we need to extrapolate our MAP numbers on three extant concept sets to some "imaginary" concept sets with larger sizes under the assumption that the additional concepts have a similar quality to the existing concepts. The first step for extrapolation is to determine a reasonable extrapolation function for the given points in order to determine the relationship between MAP and the size of concept set. Since we only have three points to fit, theoretically there are infinite amount of functions that can be used to model the given numbers. However, we can impose a reasonable assumption upon the function space so as to determine a unique extrapolation function, i.e., the maximum MAP increment brought by a new concept is proportional to the difference between the current MAP and the upper limit 1. In other word, it means that the higher the current MAP is, the less benefit a new concept can offer. According to this assumption, we can come up with the following partial differential equation,

$$\frac{dm}{dx} \propto (1 - m),$$

where m is the value of MAP, x is the number of concepts, and the boundary condition is $m(\infty) = 1$. By solving this equation, it yields,

$$m(x) = 1 - \exp(ax + b),$$

where a, b are two arbitrary numbers which can be determined by curve fitting approaches.

Figure 3.12 plots the true MAPs on three concepts sets as well as the fitted curves using the proposed exponential function over both perfect concepts and noisy concepts under oracle combination, as well as the noisy detection at 10% breakeven precision/recall and realistic combination, which assumes a 50% degradation in the combination step. It shows that the fitted curves provide a fairly accurate estimation on the target points. These curves also allow us to study the behavior of video retrieval system in depth if there are many more concepts available than the current status quo. To reach the level of a "good" retrieval system, we will only need around 800 concepts if we can obtain perfect detection outputs for each concept, and more realistically, if overall accuracy of concept detection is only 10%, we will probably need more concepts to achieve the same level, i.e., around 1,200 to 1,300 concepts. From above results, we can conclude that a couple thousand concepts should be sufficient to cover the most crucial contents in video corpora and provide a foundation to construct good video retrieval systems. Moreover, we also realize that there is a trade-off between the minimal number of concepts and the concept detection accuracy. The lower the detection accuracy, the more concepts we need to cover the video content.

Somewhat surprisingly, Figure 3.12 shows that when using 3000 or more concepts, even a fairly low detection accuracy of any single concept will be sufficient, even in the realistic combination scenario, which tries to approximate the combination behavior of real systems when compared to oracle combination. While both the number of high-level concepts and concept detection accuracy needs to be taken into account when designing the video retrieval systems, this data suggests that it is better to add more concepts into the mix rather than building very accurate concept detectors, as long as we can find a reasonable method to combine them. We also want to admit that the discounting factor between realistic combination and oracle combination is also dependent on the choice of concept combination methods. If the combination method has to rely on the semantic meaning of the concepts, its retrieval performance usually suffers more than a method that ignores explicit concept semantics, especially when the detection accuracy is as low as 10%. This suggests us to consider the semantic-insensitive and learning-based combination methods when the concept detection accuracy is not high enough.

3.4 Conclusion

In this chapter, we describe and compare a wide range of ranking feature extraction approaches for multimedia data. The first part of this chapter has described and compared the state-of-the-art text retrieval and image retrieval approaches in multimedia retrieval systems. Numerous factors of text/image retrieval have been discussed in detail, including retrieval models, text sources, expansion window size, query expansion, visual features, similarity measures and their combination strategies. The retrieval performance with various settings are evaluated in TRECVID video collections. Our experiments have confirmed the following conclusions,

1. In text retrieval, Okapi models usually provide better performance than vector space models, as is similar to previous experiments in pure-text collections. However unlike text collections, text retrieval in video corpus is relatively insensitive to the choice of document length normalization schemes. Among five predefined query types, text retrieval are most effective in the queries of finding persons and specific objects. Among all available text sources, closed caption provides the best performance, but speech transcript also achieves comparable results in term of average precision even with a word error rate around 20%. VOCR is shown to be useful in person-finding queries but not in others. Putting all text sources together is often superior to using any single source except for some rare cases. Expanding the text retrieval results to neighbor shots is an important strategy to mitigate the issue of timing misalignment between video clips and relevant text keywords. But for news video retrieval, it is beneficial to limit the expansion size inside the story boundary. Manual query expansion with a careful keyword selection can considerably improve text retrieval performance, especially for the queries related to sports events. But on the other hand, automatic query expansion based on WordNet or local feedback might degrade the retrieval performance if it is not handled properly.

2. In image retrieval, color-based features (especially color moment features) are one of the best choices with high effectiveness and low computational

cost. The edge histogram can occasionally provide a comparable performance with color-based features but its performance is not as consistent. With respect to each query type, image retrieval is particularly useful for specific object type and sport type queries, of which the information need can be captured by the visual appearance of a limited number of image examples. But it produces relatively poor performance on the other query types. Being robust to outliers and efficient to compute, the L_1 distance is shown to be one of the most effective distance metrics in practice. To combine multiple query images, using the harmonic mean and maximum function outperforms the other fusion functions in terms of mean average precision. Combining the outputs of text retrieval and image retrieval can consistently improve the retrieval performance based on any single modality. Meanwhile, it is more robust and effective if the combination is done in a query-dependent way.

In the second part of the chapter, we presented the general approaches and discussed several open research directions to detect semantic concepts in multimedia collections. These concepts can be seen as an intermediate step in enabling video search and retrieval. Typical semantic concept detection approaches are made up of four major steps: manual annotation, low-level feature extraction, supervised learning and multi-modal fusion. The successes of previous approaches have demonstrated that a large number of high-level semantic concepts can be directly inferred from low-level multi-modal features without demanding a prohibitive amount of human annotation efforts. But several open research directions still need to be explored, such as balancing training distribution, leveraging unlabeled data, modeling relationship between concepts and extracting hidden concepts from video collections. Moreover, we contribute several novel approaches for semantic concept detection, e.g., SVM ensembles to handle rare class, semi-supervised cross feature learning to leverage multimodal information, undirected graphical models to model concept relations and dual-wing harmoniums to discover hidden concepts. Finally, our case study results confirmed that a few thousand semantic concepts could be sufficient to support high accuracy video retrieval systems. When sufficiently many concepts are used, even low detection accuracy can potentially provide good retrieval results as long as we find a reasonable way to combine them.

Chapter 4

Basic Probabilistic Model for Multimedia Retrieval

In this chapter, we proposed using a relevance-based probabilistic retrieval model as a principled framework to combine diverse knowledge sources in multimedia retrieval. For parameter estimation, we discuss a discriminative learning approach to learn the combination weights from training data, followed by an extensive discussion on selecting appropriate features and computing the limits of using linear combination. In order to incorporate ranking information into the learning process, an efficient rank learning approach called "ranking logistic regression" has been developed that can explicitly model the ranking relations with much less training time. Finally, we conclude the chapter with discussing some related work on weight estimation and rank learning.

4.1 Notations and Terminologies

We begin by introducing the notations and terminologies used in this work. The term *multimedia document* or *document* is referred to as the basic unit of retrieval throughout this proposal. For example, in the TRECVID video retrieval task, the multimedia documents stands for video shots, i.e., video clips from a single continuous camera operation without an editor's cut, fade or dissolve. We define the other notations as follows,

- A query collection $\mathcal{Q}$ contains a set of queries: $\mathcal{Q} = \{Q_1, ..., Q_t, ..., Q_{M_Q}\}$ where Q_t has text descriptions and possibly image/audio/video query examples. By default, we denote Q as the current query submitted by the user;

- A multimedia collection $\mathcal{D}$ contains a set of multimedia documents: $\mathcal{D} = \{D_1, ..., D_j, ..., D_{M_D}\}$ where each document consists of low-level features from multiple modalities. D_q^+ is the collection of relevant documents and D_q^- is the collection of irrelevant documents for query q. M_D^+ and M_D^- are the number of documents in each collection;

- Each document D_j is represented as a bag of ranking features $\{f_1, ..., f_N\}$ generated from different knowledge sources. To be more precise, the document D_j can be represented as a vector of ranking features based on the following two types of ranking features: 1) N_r retrieval experts from different modalities, i.e., $f_i^r(D_j, Q)$ given the query Q, and 2) N_s semantic video concepts independent to the query, i.e., $f_i^s(D_j)$, where $N_r + N_s = N$. In the rest of our discussions, we will simplify the notations of both types of outputs to be $f_i(D_j, Q)$.

- Retrieval can be regarded as a binary classification problem, of which the goal is to estimate the conditional probability of relevance. Let $y \in \{-1, 1\}$ indicates the document D is relevant or irrelevant to the query Q. Given a pair of document D and query Q, we denote the conditional probability of relevance as $P(y = 1|D, Q)$, or equally $P(y_+|D, Q)$. Similarly, we denote the conditional probability of irrelevance as $P(y = -1|D, Q)$ or $P(y_-|D, Q)$. In some cases, we will alternatively use the indicator in another form where $y' \in \{0, 1\}$, where $y' = (y + 1)/2$.

Under this representation, the multimedia retrieval problem can be formalized as follows: *Given a query Q, a document D_j, and their ranking features $f_i(D_j, Q)$, estimate a combination function of $F(Q, D_j, f_1, ..., f_N)$ to predict the conditional probability of relevance $P(y_+|D_j, Q)$.*

4.2 Conditional Probabilistic Retrieval Models

For the text-based retrieval, conventional relevance-based probabilistic models [Fuh92] rank documents by sorting the conditional probability that each document would be judged relevant to the given query, i.e., $P(y_+|D, Q)$. Typically, a document is described by a vector of word indices or probabilistic indexings $(t_1, ..., t_n)$.

The underlying principle using probabilistic models for information retrieval is called *Principal Ranking Principle* [Rob77] that suggests sorting the documents D by the log-odds of document relevance, where the odds ratio $O(y|D,Q)$ is defined as $\frac{P(y_+|D,Q)}{P(y_-|D,Q)}$. There exist a number of models to estimate $O(y|D,Q)$ in the literature and most of them have a root from the *Binary Independence Model*(BIM) [RJ77]. This model represents each document as a binary vector of the term presence/absence with all the information of term frequencies discarded. To estimate the term weights, BIM proceeds by inverting the position of y and D based on the Bayes rule and estimating the generative probabilities of document D in the relevant and irrelevant documents,

$$O(y|D,Q) = O(y|Q)\frac{P(D|y_+,Q)}{P(D|y_-,Q)} = O(y|Q)\prod_{i=1}^{N}\left(\frac{p_i}{r_i}\right)^{t_i}\left(\frac{1-p_i}{1-r_i}\right)^{1-t_i} \quad (4.1)$$

where t_i is the binary indexing for the i^{th} term t_i, p_i is the generative probability given relevant documents $P(t_i = 1|y_+)$ and r_i is the generative probability given irrelevant documents $P(t_i = 1|y_-)$. The last step of Eqn(4.1) can be derived under the independence assumption between retrieved documents and the independence assumption between indexed terms. By taking the logarithm on both sides, the decision function can be transformed into a linear function of t_i,

$$\log O(y|D,Q) = w_0(Q) + \sum_{i=1}^{N} t_i \log\frac{p_i(1-r_i)}{q_i(1-r_i)} = w_0(Q) + \sum_{i=1}^{N} t_i w_i(Q), \quad (4.2)$$

where $w_0(Q)$ is the constant term corresponding to $\log O(y|Q)$, $w_i(Q)$ is the term weight $\log\frac{p_i(1-r_i)}{r_i(1-p_i)}$ for the i^{th} word. We represent w_i as a function of the query Q in order to emphasize the fact that the term weights are query specific parameters. Note that the log-odd is an attractive measure for the retrieval model in the sense that term weights are accumulated additively and the entire scale from $-\infty$ to ∞ can be used. Therefore, Eqn(4.2) allows the term weights being set in any scale without restrictions.

However, the binary representation might be too restrictive for term presence. Therefore, researchers further refine the binary independence model by extending binary term indexing to be probabilistic indexing that indicates how accurately a term is assigned to the documents. Formally, we replace the term index t_i in Eqn(4.1) to be the conditional probability of t_i given the query and document, i.e., $P(t_i|D,Q)$ and assume $P(t_0|D,Q)$ is a constant of 1. Then we can obtain the following model,

$$\log O(y|D,Q) = \sum_{i=0}^{N} w_i(Q)P(t_i|D,Q). \tag{4.3}$$

To model the multimedia retrieval problem in a similar framework as Eqn(4.3), we can substitute the i^{th} probabilistic word indexing $P(t_i|D,Q)$ and the term weight w_i with the i^{th} indexing of ranking features $f_i(D,Q)$ and the combination weight λ_i respectively in the above formula, i.e.,

$$\log O(y|D,Q) = \sum_{i=0}^{N} \lambda_i(Q)f_i(D,Q). \tag{4.4}$$

In this chapter, we simply assume the combination weight $\lambda_i(Q)$ is independent of the query Q and rewrite the retrieval model as follows,

$$\log O(y|D,Q) = \sum_{i=0}^{N} \lambda_i f_i(D,Q) = \lambda_0 + \sum_{i=1}^{N_r} \lambda_i f_i^r(D,Q) + \sum_{i=1}^{N_s} \lambda_{(i+N_r)} f_i^s(D).$$

or equivalently,

$$P(y_+|D,Q) = \sigma\left(\sum_{i=0}^{N} \lambda_i f_i(D,Q)\right) = \left[1+\exp\left(-\sum_{i=0}^{N} \lambda_i f_i(D,Q)\right)\right]^{-1}, \tag{4.5}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the standard logistic function and λ_i is the combination parameter for the i^{th} ranking feature $f_i(D,Q)$. Eqn(4.5) summarizes the basic multimedia retrieval framework which naturally provides a probabilistic interpretation for the retrieval outputs. In this model, we can find that the decision of relevance is determined by the weighted linear combination of ranking features . Finally, according to the *Principal Ranking Principle*, the documents are presented to users in the descending order of $P(y_+|D,Q)$.

To summarize, we present a general relevance-based probabilistic retrieval model for multimedia retrieval. As can be seen later, this model is able to be extended to incorporate various useful factors into the retrieval task such as query information, context information and human feedback. In the previous chapter, we already describe how various ranking features can be generated from multimedia documents. But before we proceed, another major issue has yet to be resolved, i.e., how we can provide effective methods to estimate the combination parameters λ_i. This issue is discussed in the following sections.

Remark: Interestingly, the proposed retrieval model can be interpreted from the perspective of the vector space model. Let us consider the scenario when we have one ranking feature generated from text retrieval and N_s other ranking features indexed by high-level semantic concepts, we can rewrite Eqn(4.4) to be,

$$\log O(y|D,Q) =_r \lambda_1(Q) f_1^r(D,Q) + \sum_{i=1}^{N_s} \lambda_{i+1}(Q) f_i^s(D), \tag{4.6}$$

where $=_r$ means equivalence in ranking. If we further assume the $f_1^r(D,Q)$ to be the text retrieval output from a vector space model, say, the model with SMART code $nnn.nnn$ which means raw term frequency(tf) are used to encode both queries and documents, Eqn(4.4) can be updated as,

$$\log O(y|D,Q) =_r \sum_{j=1}^{W} \lambda_1(Q) Q_j^{tf} D_j^{tf} + \sum_{i=1}^{N_s} \lambda_{i+1}(Q) f_i^s(D), \tag{4.7}$$

where W is the number of indexed term in the vocabulary. As can be observed, the indexing of i^{th} semantic concept can alternatively be viewed as a special "tagging" word with the query tf encoded in $\lambda_i(Q)$ and the document tf encoded in $f_i^s(D)$. In some sense, we are building up an expanded vocabulary by incorporating the additional semantic concepts into the original set of indexed words.

However, unlike the canonical vector space model, estimating the query tf for semantic concepts, i.e., $\lambda_i(Q)$, is not that straightforward. To achieve this, one way is to extract the same number of semantic concepts for queries and view their outputs as query tf. But since a lot of semantic concepts are predicted based on visual features, it could be difficult to process queries with the same detection models, because it might require the existence of image examples and a long processing time. Another workaround is to match the name of each concept with the query terms. If a concept name is found to be part of the query description, then $\lambda_i(Q)$ can set to be a positive value. For example, the $\lambda_i(Q)$ of the semantic concept of "building" detection will be a positive value for retrieving the query of "finding the scenes containing buildings in New York city". However, in practice it is unrealistic to expect a general user to explicitly indicate all related concepts in his query description. Therefore, in this thesis, we develop a series of advanced retrieval models to provide alterative ways to estimate the query tf.

The proposed model also has connections with the language modeling approaches. In fact, Lafferty et al. [LZ03] pointed out that language modeling and relevance-based probabilistic models are actually two sides of the same coin. For language modeling approaches, relevance is an implicit variable in the modeling process so that what we are estimating is $P(Q|D, y_+)$ rather than $P(Q|D)$. To show this, we can rewrite Eqn(4.4) to be,

$$\log O(y|D, Q) = \log \frac{P(Q|D, y_+)}{P(Q|D, y_-)} + \log \frac{P(y_+|D)}{P(y_-|D)} =_r \sum_{i=1}^{N} \lambda_i(Q) f_i(D, Q). \quad (4.8)$$

By assuming $P(Q|D, y_-) = P(Q|y_-)$ [LZ03] and denoting $f_0(D) = \log \frac{P(y_+|D)}{P(y_-|D)}$ which is an query-independent feature, we can have

$$P(Q|D, y_+) =_r \exp \left[f_0(D) + \sum_{i=1}^{N} \lambda_i(Q) f_i(D, Q) \right] \quad (4.9)$$

If the underlying retrieval experts (text/image retrieval) are all built upon the language modeling approaches [Wes04] and their ranking features are defined in the logarithm domain $f_i^r(D, Q) = \log P_i(Q|D)$ where $P_i(Q|D)$ denotes the language model learned from each modality, it yields

$$P(Q|D, y_+) =_r \prod_{i=1}^{N_r} P_i(Q|D)^{\lambda_i(Q)} \cdot \exp \left[f_0(D) + \sum_{i=1}^{N_s} \lambda_{(i+N_r)}(Q) f_i^s(D) \right] \quad (4.10)$$

As we can observe, $P(Q|D, y_+)$ can be decomposed into the products of the discounted probability $P_i(Q|D)^{\lambda_i(Q)}$ generated from the language model on each modality, the weighted output of semantic concepts $e^{\lambda_{(i+N_r)} f_i^s(D)}$ and a document importance term $e^{f_0(D)}$. This analysis provides an intuitive explanation of the proposed model from the aspect of the language modeling approach.

The proposed retrieval model still bears some limitations. For instance, ranking features are combined using a linear function and thus it cannot handle the case of non-linear combination. Also, we assume that knowledge sources are independent to each other but in fact they are not. How to augment the current retrieval model and whether the advanced models will help remain to be a open research topic for us.

4.3 Parameter Estimation

With all the ranking features available, the following step in the probabilistic retrieval model is to estimate the corresponding combination parameters λ_i. These parameters can be estimated in a variety of ways. The main approach in the early years is to learn the parameters based on user relevance feedback. But this approach is greatly limited by the requirement of additional user annotations. More recent approaches try to estimate w_i by making simplifying assumptions on the class-conditional distributions. It can be shown that under some specific assumptions, the relevance-based probabilistic model can be mapped to a vector space model with *TF*IDF* weighting [CH88]. However, despite their popularity and successfulness, the underlying model assumptions of these approaches such as the term independency are not always correct in practice. In contrast, discriminative learning approaches can directly model the classification boundary and typically make fewer model assumptions. They have been applied in many domains of text processing such as text classification and information extraction. Moreover, different retrieval sources usually provide heterogeneous types of outputs for combination including both query-dependent features and query-independent features. Generative learning might face difficulties in the manual design of different model distributions for these outputs. Nallapati [Nal04] has shown that in presence of heterogenous features, discriminative models are superior to generative models in a home-page finding task. Moreover, since the number of retrieval sources is usually much smaller than the number of text keywords, it allows a discriminative model to estimate the parameters more robustly given the same amount of the training data.

After taking multiple factors into account, we decide to adopt discriminative learning approaches to estimate the parameters which can directly model the classification boundary and benefit from requiring less model assumptions. Among various discriminative models, logistic regression is a natural choice for learning the probability when the log-odd is a linear function [Gey94]. Therefore, we consider using logistic regression for weight estimation with some manually collected training data. The graphical representation for this learning model is shown in Figure 4.1. Formally, given the relevance judgment y_{tj} for the training queries Q_t and documents D_j, the maximum-likelihood estimation for the combination parameters λ is as follows,

$$\begin{aligned}\lambda^* &= \arg\max_{\lambda} \prod_{t=1}^{M_Q} \prod_{j=1}^{M_D} P(y_+|D_j, Q_t)^{y'_{tj}} (1 - P(y_+|D_j, Q_t))^{1-y'_{tj}} \\ &= \arg\max_{\lambda} \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} \log \left[\sigma \left(y_{tj} \sum_{i=1}^{N} \lambda_i f_i(D_j, Q_t) \right) \right]. \qquad (4.11)\end{aligned}$$

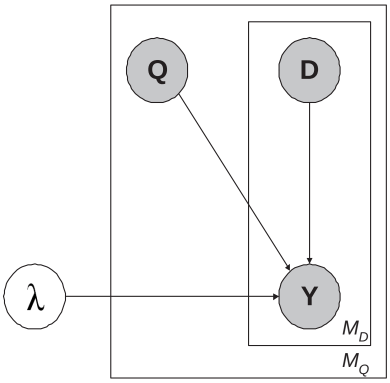


Figure 4.1: Graphical model representation for the conditional probabilistic retrieval model. The nodes with known values are shaded and others are unshaded. The plates stand for replicas of the subgraphs where the number of replicas is on the corner.

The maximum likelihood estimator λ^* has to be found numerically. Various numerical optimization algorithms, e.g., Newton's method, conjugate gradient and iterative scaling, can be applied to optimize the log-likelihood of logistic regression. In the implementation, we adopt a well-known optimization algorithm called iterative reweighted least squares and its learning process is shown in Figure 4.2.

4.3.1 Feature Selection

Typically, learning algorithms treat every ranking feature as equally predictive if no prior knowledge are available. However in practice, the predictive power of many ranking features is only limited to a certain subset of domains. For example, the high-level semantic concept of "Rocket Launching" can only be useful for the query related to rocket/missiles but not as predictive in other queries. Therefore we categorize the set of ranking features into two types, i.e., general and domain-specific features. The general features have universal predictive powers for large number of queries, such as anchor detection, commercial detection and face detection. On the other hand, the domain-specific features such as "Rocket Launching" are only useful for a limited number of queries. Christel et al. [CH05] investigated the distinctions between general and query-specific semantic concepts. According to above analysis, we need to apply some feature selection method to choose general ranking features before running the learning algorithm. In our experiments, we apply a χ^2 test [YP97] to select the subset of general ranking features. The χ^2

Input: Feature matrix $F^{M_D \times N}$ where $F_{ij} = f_i(D_j, Q)$ and $F_j = (f_{1j}, ..., f_{M_D j})$.
Document relevance $Y = (Y_1, ..., Y_{M_D})^T$.
Output: Weights $\lambda = (\lambda_1, ..., \lambda_N)^T$ that maximizes the log-likelihood.
Algorithm:
Let $\lambda^{(1)}$ to be the zero vector. For $k = 1, 2, ...$

1. Compute the fitted value

$$\pi_j = \frac{\exp(F_j \lambda^{(k)})}{1 + \exp(F_j \lambda^{(k)})}, j = 1, ..., M_D$$

2. Define an $n \times n$ weight matrix W whose i^{th} diagonal element is $\pi_j(1 - \pi_j)$

3. Define the adjusted response vector

$$Z = F\lambda^{(k)} + W^{-1}(Y - \pi),$$

   where $\pi = (\pi_1, ..., \pi_{M_D})$

4. Take the weighted linear regression of Z on X,

$$\lambda^{(k+1)} = (F^T W F)^{-1} F^T W Z.$$

   The standard errors are given by $V(\lambda) = (F^T W F)^{-1}$.

Figure 4.2: The reweighted least square algorithm for logistic regression

statistics is generally computed to measure the dependence of two random variables. In our case we use it to measure the dependence between each ranking feature and the relevance judgment. If a ranking feature is suggested to be independent to the document relevance variable (over all queries), this feature will be treated as query-specific and be eliminated in the learning process.[1] By doing this, we also enjoy the advantage of significantly reducing the feature set size and the computational time in the learning process.

[1] The features with continuous probabilistic outputs will be converted to discrete output with a threshold of 0.5.

4.3.2 The Limits of Linear Combination

Multimedia retrieval can benefit from combining multiple knowledge sources, however, several open questions remain to be addressed: what are the limits of these combination methods if we already have scores for each different component? Is linear combination sufficient or are more complex functions necessary? Is it sufficient to assign equal weights for the ranking features in all queries? In our previous work [YH03], we developed a theoretical framework for studying the performance limits over both monotonic and linear combination functions assuming the retrieval results from every media component are known. This analysis may give us some answers to the above questions and help in finding ways to boost the performance of all multimedia retrieval systems.

Here we only present the approach to compute the optimal linear combination performance with the ranking features f_i fixed. Let us denote $F(D, Q) = \sum_{i=1}^{N} \lambda_i f_i(D, Q)$ and $AP(F(D, Q))$ the average precision of order list σ where σ is determined by retrieval score $F(D, Q)$ with respect to D. Therefore our task can be rewritten as a bound constrained global optimization problem,

$$LLB = \max_{\lambda_i} AP\left(\sum_{i=1}^{k} \lambda_i f_i(D, Q)\right),$$

which we call LLB as locally fixed linear bound. Note that this bound assumes that users can assign unequal weights for different queries. If we pose an additional constraint, namely that the λ_i cannot be changed across queries, this might give a more tighter bound on the performance, which we call globally fixed linear bound(GLB).

To handle the bounded constraint optimization problem, we use a global optimization algorithm called MCS algorithm as proposed by Huyer et al [HN99]. Their method combines both global search and local search into one unified framework via multi-level coordinate search. It is guaranteed to converge if the function is continuous. Also in our implementation, multiple start points have been tried for each query to avoid the local minima problems.

4.4 Beyond Binary Classification: Ranking Logistic Regression

The aforementioned retrieval models cast information retrieval into a binary classification problem that treats the relevant query-document pairs as positive data and irrelevant pairs as negative data. Despite its great successfulness, converting retrieval into classification might suffer from several disadvantages. For example, since the classification accuracy has no direct relationship with the retrieval measure, a learning algorithm that can achieves a high classification accuracy might not produce a good performance in terms of ranking. Therefore, there are a few recent attempts to develop learning algorithms that can explicitly account for ranking relations in information retrieval [Joa02, FISS98, Bea05, CNG$^+$05, GQXN05]. Most of these rank learning approaches attempt to model the pairwise ranking preferences between every pair of relevant and irrelevant training examples. They are built on a solid foundation because it has been shown that minimizing the discordant pairs of examples are closely related to the commonly used ranking criteria. However, the effort of modeling every pair of examples often leads to a prohibitive learning process and thus limits their applications in practice.

In this section, we propose a general rank learning framework based on the principle of margin-based risk minimization, which can be generalized to a large family of rank learning approaches such as Ranking SVMs [Joa02] and RankBoost [FISS98]. To make the optimization less computationally intensive but still keep the ability to model the ranking relations between examples, we further propose an approximate but efficient rank learning framework by approximating the pairwise risk function. In particular, we designed a new learning algorithm called ranking logistic regression(RLR) by plugging in the logit loss function, which has a similar form as classical logistic regression except the positive data are weighted stronger to balance the positive/negative data distribution and meanwhile the median value of each feature is shifted to zero. This algorithm serves as the basic learning framework for the model extension in the following sections.

4.4.1 A Margin-based Framework for Learning Ranks

In information retrieval, most of the learning approaches simplify the rank learning to be a binary classification problem and many of them can be derived from a learning framework that aims at minimizing the following regularized empirical risk [HTF01],[2]

$$\min_f R_{reg}(f) = \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} L(y_j f(d_j, q_t)) + \nu \Omega(\|f\|), \tag{4.12}$$

[2] For example, logistic regression can be viewed as a special case with a logit loss function.

where y_j is the binary relevance label for j^{th} training document d_j, L is the empirical loss function, $\Omega(\cdot)$ is some monotonically increasing regularization function, $\|f\|$ is the norm of function f and ν is the regularization constant. The component of $yf(d,q)$ is usually called "margin" in the literature [HTF01] and hence the learning framework is called *margin-based risk minimization* framework. However, such a classification framework might have difficulties in dealing with the retrieval task. For example, because there are only a small fraction of relevant examples in the collection and many others are left as irrelevant ones, a classification algorithm that always provides negative prediction will unfortunately achieve a high predictive accuracy. Moreover, the classification accuracy has no relationship with the retrieval measure such as average precision. Maximizing the classification accuracy does not necessarily imply a higher ranking effectiveness. To address this issue, we can consider switching the learning criterion to optimize the number of discordant pairs Q between the predicted ranking and the target ranking, i.e., $\sum_{q_t} \sum_{d_j \in D^+_{qt}} \sum_{d_k \in D^-_{qt}} I(f(d_j,q_t) - f(d_k,q_t))$ where $I(\cdot)$ is the indicator function. Unfortunately, a direct optimization on the general form of above equation has been shown to be NP-hard. But following an idea similar to margin-based risk minimization, we can replace the binary misclassification error $I(\cdot)$ into a continuous, convex and monotonically decreasing loss function $L(\cdot)$ in an attempt to facilitate the learning process. By introducing an additional regularization term, we can obtain the following unified margin-based rank learning framework,

$$\begin{aligned}\min_f RR_{rank}(f) &= \sum_{q_t \in Q} \sum_{d_j \in D^+_{qt}} \sum_{d_k \in D^-_{qt}} L(f(d_j,q_t) - f(d_k,q_t)) + \nu\Omega(\|f\|) \\ &= \sum_{q_t \in Q} \sum_{d_j \in D^+_{qt}} \sum_{d_k \in D^-_{qt}} L\left(\sum_{i=1}^{n} \lambda_i [f_i(d_j,q_t) - f_i(d_k,q_t)]\right) + \nu\Omega(\|f\|), \quad (4.13)\end{aligned}$$

where the retrieval function $f(d_j,q)$ is expressed as a linear function of the ranking features due to its retrieval effectiveness and simple presentation, i.e., $f(d_j,q) = \sum_{i=1}^{n} \lambda_i f_i(d_j,q)$. By optimizing the risk function, we can compute the risk minimization estimator λ_i^* for each ranking feature $f_i(d_j,q)$. With different choices of loss functions and regularization terms, a large family of rank learning algorithms can be derived from Eqn(4.13). For example, ranking support vector machines can be obtained by setting loss function to be the hinge loss and regularization factor to be $\|w\|_2^2$. RankBoost can be viewed as a rank learning algorithm with the exponential loss function. A recent proposed linear discriminant ranking model(LDM) [GQXN05] can be derived by using a binary loss function without regularization terms and setting $f(d_j, q)$ to be a linear function.

The rank learning framework presented in Eqn(4.13) lends itself to another advantage over the margin-based classification framework. Before further discussions, let us define a useful property called rank consistency,

Definition 4. ***(Rank consistency)*** *If a risk minimization estimator λ_i^* satisfies the following conditions: 1) $\lambda_i^* \geq 0$ when $\forall d_j \in D_q^+, \forall d_k \in D_q^-, f_i(d_j, q) \geq f_i(d_k, q)$, and similarly 2) $\lambda_i^* \leq 0$ when $f_i(d_j, q) \leq f_i(d_k, q)$, we will call the estimator is consistent with the data ranking. Note that we assume $f_i(\cdot)$ does not take a trivial constant value.*

It is intuitive to expect the parameters estimated from a rank learning algorithm to satisfy the property of rank consistency. For instance, let us assume the binary outputs of an anchor person detector is one of the ranking features in the multimedia retrieval system, where $f_a = 0$ means no anchor available and $f_a = 1$ otherwise. For a specific query, if we find all of the relevant documents do not contain any anchor shots, i.e., $f_a(d_j, q) \leq f_a(d_k, q)$, then it is naturally to expect the corresponding weight λ_a to be lower than 0, because a negative λ_a can push the relevant examples closer to the top ranked positions.

Unfortunately, simple margin-based classifiers do not offer any guarantees on this intuitive property. In other words, for a ranking feature, even when its values in the relevant documents are always lower than that in the irrelevant documents, the corresponding weight estimator can still be positive. This is because general classification algorithms did not take the ranking information into consideration and the violation of rank consistency might sometimes provide better separability between positive/negative examples rather than better retrieval performance. In contrast, the proposed margin-based rank learning framework preserves such a property, namely, the λ^* learned from Eqn(4.13) is always consistent with the data ranking if $L(\cdot)$ is a monotonically decreasing function. The proof is given in Appendix. This fact further explains why the proposed margin-based rank learning framework is a better candidate for the retrieval problem.

4.4.2 Ranking Logistic Regression

The above margin-based rank learning framework is quite general, but as mentioned before, optimizing the pairwise risk function in Eqn(4.13) in a brute force manner needs to take care of an explosive number of training pairs between every relevant and irrelevant documents. Therefore, it is desirable to develop a more efficient algorithm to speed up the learning process. In this section, we will describe one such efficient learning algorithm derived from the general rank learning framework. Unless stated otherwise, the following discussions assume the loss function $L(\cdot)$ is convex and satisfies $2L(x/2) \geq L(x)$. Under this assumption[3], we can have the following inequality,

$$RR_{prox}(f) \geq RR'_{rank}(f) \geq \frac{1}{2}[RR_{prox}(f) - RR_{prox}(-f)], \tag{4.14}$$

where $RR'_{rank}(f)$ is the pairwise ranking risk defined in Eqn(4.13) without the regularization factor and $RR_{prox}(f)$ is the approximate ranking risk function based on a shifted retrieval function $f^{\alpha}(d_j, q) = \sum_{i=1}^{n} \lambda_i [f_i(d_j, q) - \alpha_i]$,

$$RR_{prox}(f) = \sum_{q_t} \left\{ \sum_{d_j \in D_{qt}^+} M_D^- L\left(f^{\alpha}(d_j, q_t)\right) + \sum_{d_k \in D_{qt}^-} M_D^+ L\left(-f^{\alpha}(d_k, q_t)\right) \right\}.$$

The proof of inequality Eqn(4.14) is provided in the Appendix. Both bounds are tight in the sense that all three parts are equal when $L(\cdot)$ is a linear function. Therefore, in lieu of optimizing the pairwise ranking function, we can consider minimizing the $RR_{prox}(f)$ as a reasonable surrogate. Meanwhile, it is instructive to compare and contrast $RR_{prox}(f)$ with the margin-based classification risk function presented in Eqn(4.12). As can be seen, if we set the label y of d_j to be +1 and that of d_k to be -1 in Eqn(4.12), these two risk functions have a similar form with each other. Therefore, minimizing $RR_{prox}(f)$ has a small computational complexity $O(M_D^+ + M_D^-)$, which is much faster than minimizing the pairwise ranking risk function with a complexity $O(M_D^+ M_D^-)$. However, $RR_{prox}(f)$ also bears some major differences with the classification risk R_{reg}, because (1) it weights the relevant documents more heavily by a ratio of M_D^-/M_D^+; (2) it drops the constant feature term, which is usually available for classification to capture the shifts of decision boundary; 3) it shifts each feature vector by the parameter α_i. These differences have made the $RR_{prox}(f)$ a better choice for the rank learning, with the advantage of balanced data distributions.

In the following implementation, we specially adopt the logit loss $L_R(x) = -\log \sigma(x) = \log(1 + \exp(-x))$ as the empirical loss function. But before proceeding we need to decide the value of the shifting parameters α. One idea is choose α to minimize the gaps between the lower bound and the upper bound, i.e.,

[3] This is a very general condition with a large family of loss functions satisfied, such as the hinge loss(SVMs), logistic loss and binary loss function.

$\min_\alpha[RR_{prox}(f) + RR_{prox}(-f)]/2$ so as to make RR_{prox} a tight approximation for RR'_{reg}. We approach this by utilizing the inequality $L_R(x)+L_R(-x) \leq 2+|x|$ and thus we can transform the optimization problem into a series of minimization problem with respect to each α_i,

$$\min_{\alpha_i} \sum_{q_t} \left\{ \sum_{d_j \in D^+_{qt}} M^-_D |f_i(d_j, q_t) - \alpha_i| + \sum_{d_k \in D^-_{qt}} M^+_D |f_i(d_k, q_t) - \alpha_i| \right\}. \quad (4.15)$$

The optimal estimator α^*_i can be written as follows,

$$\alpha^*_i = \text{median} \left[\bigcup_{\forall j,t} \left\{ f_i(d_j, q_t) \right\}_{M^-_D} \cup \bigcup_{\forall k,t} \left\{ f_i(d_k, q_t) \right\}_{M^+_D} \right], \quad (4.16)$$

where $\{x\}_n$ denote a set of n elements with the same value x. By substituting the optimal α^*_i and the logit loss, we can proceed to optimize the combination parameter λ^*_i as follows,

$$\max_{\lambda} \sum_{q_t} \left\{ \sum_{d_j \in D^+_{qt}} M^-_D \log \sigma \left(\sum_i \lambda_i f^*_{ijt} \right) + \sum_{d_k \in D^-_{qt}} M^+_D \log \sigma \left(- \sum_i \lambda_i f^*_{ikt} \right) \right\}, \quad (4.17)$$

where $f^*_{ijt} = f_i(d_j, q_t) - \alpha^*_i$. The optimal estimation of λ_i can be achieved by using any gradient descent methods such as iterative reweighted least squares(IRLS) algorithm [JJ94]. We prove in the Appendix that this estimation is consistent with the data ranking. In the rest of the paper, we will call this algorithm *ranking logistic regression*(RLR).

4.5 Experiments

Our experiments are designed based on the guidelines of the manual retrieval task in TREC video retrieval evaluation(TRECVID), which requires an automatic video retrieval system to search relevant documents without any human feedbacks. To evaluate the proposed learning algorithms, we used TRECVID'03-'05 video collections which officially provide 25 multimodal queries and around 70,000 shots every year[4]. Each of these video collections is split into a development set and a search set chronologically by source. For each query topic, the relevance judgment on the search set was provided officially by NIST and the judgment on

[4] Information about these collections can be found at the TRECVID web site [SO03].

the development set was collaboratively collected by several human annotators using the Informedia client [HCC$^+$04]. Although we cannot guarantee all the relevant shots can be found in the development set, this collection effort generally provides a high coverage for the relevance data based on our experience. As the building blocks of the retrieval task, we generated a number of ranking features on the search set including 14 high-level semantic features learned from development data (face, anchor, commercial, studio, graphics, weather, sports, outdoor, person, crowd, road, car, building, motion), and 5 uni-modal retrieval experts (text retrieval, face recognition, image-based retrieval based on color, texture and edge histograms). The detailed descriptions on the feature generation can be found in [HCC$^+$04].

Data	**Algorithms**	**AvgPrec**	**Prec10**	**Prec30**	**Prec100**
t05	LR	**0.207**	0.506	0.433	0.341
	SVM-L	0.202	0.507	0.433	0.338
	SVM-R	0.194	0.502	0.419	0.322
	NB	0.200	0.535	0.410	0.334
	LDA	0.197	0.525	0.442	0.333
t04	LR	0.132	0.288	0.241	0.184
	SVM-L	**0.147**	0.325	0.283	0.210
	SVM-R	**0.147**	0.358	0.283	0.208
	NB	0.129	0.231	0.215	0.182
	LDA	0.135	0.292	0.225	0.184
t03	LR	**0.185**	0.431	0.358	0.229
	SVM-L	0.183	0.425	0.360	0.223
	SVM-R	0.178	0.431	0.352	0.222
	NB	0.181	0.344	0.338	0.230
	LDA	0.182	0.406	0.358	0.228

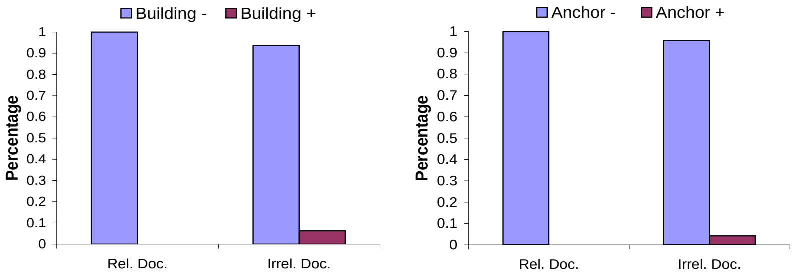
Table 4.1: Comparison of learning algorithms on TRECVID'03-'05 data. AP is the mean average precision on the search sets. Prec10, Prec30 and Prec100 indicate the mean precision at the top 10, 30 and 100 retrieved shots.

4.5.1 Generative vs Discriminative Models

We compare five different types of algorithms on all three video collections in Table 4.1, i.e., three discriminative models including logistic regression(LR), support vector machines with linear decision boundaries(SVM-L), support vector machines with RBF kernels(SVM-R) and two generative models including naïve Bayes(NB) and linear discriminant analysis(LDA). For each algorithm, we learned the combination weights on a per query basis using the development data. To reduce the learning complexity, we choose the top 1000 shots with the highest text retrieval scores as the training examples. The learned models are evaluated based on the same query using the search set. By averaging the performance on all queries, we report the retrieval performance in terms of the mean average precision(MAP) and precision at top 10, 30 and 100 retrieved shots. To guarantee the learning process being supported by sufficient training data, we intentionally removed the queries with less than 10 positive examples in the training process, which typically decreased the query number to around 20 for each data collection. As shown in Table 4.2, the discriminative models such as LR and SVM-L are usually superior to the generative model, i.e., NB and LDA, in terms of both the testing MAP on three collections, because the ranking features generated from different components often bear dissimilar distributions and the discriminative model has the flexibility to model the decision boundary with less model assumptions than the generative model. The performance of SVM-R exhibits inconsistent performance across various data collections, which might indicate the unstableness of non-linear classifiers for knowledge source combination due to their unnecessary model complexities.

4.5.2 Rank Learning

Figure 4.3 provides two illustrative examples from TREC'05 query topics to describe the distinctions between logistic regression(LR) and its ranking counterpart(RLR) with respect to the property of rank consistency. For example, it can be observed from the Figure 4.3(a) that none of the relevant documents in the development data are classified as building scenes while some (around 5%) of the irrelevant documents are detected as building scenes. Intuitively, a rank consistent learning algorithm should be able to estimate a negative weight corresponding to the building features since the values of building features in the relevant documents are not higher than those in the irrelevant documents. However, LR instead learned a positive weight for the building feature which can produce a larger log-



$\lambda_{LR} = +1.2\ AP = 0.23$
$\lambda_{RLR} = -19.7\ AP = 0.41$
(a) Query "roads and cars"

$\lambda_{LR} = +2.1\ AP = 0.81$
$\lambda_{RLR} = -3.3\ AP = 0.85$
(b) Query "soccer goals"

Figure 4.3: (a) Distribution of the {-1,1}-output building detection features in the (ir)relevant training documents for the query "finding roads and cars". The red bars mean how many percents of (ir)relevant documents are detected as building scene. The blue bars mean the percentage detected as non-building scene. (b) Distribution of the anchor detection features for "finding goals in a soccer game". The numbers below indicate the combination weights and the training average precision learned by LR and RLR.

likelihood and better class separability on the development data. Unfortunately, this criterion is not directly correlated to the ranking itself. In contrast, RLR reversed the sign of predicted weights and make it be rank consistent to the data. Thus, RLR achieves a better average precision than LR in the development data. As another example, Figure 4.3(b) shows the distribution of the anchor detection feature in the query "finding goals in a soccer game", from which we can come up with a similar conclusion.

We compare three different types of algorithms on all three video collections in Table 4.2, i.e., logistic regression(LR), ranking logistic regression(RLR), full ranking logistic regression(FRLR) which directly optimizes the pairwise risk function in Eqn(4.13). The experimental setting is similar as previous one. Among these models, the ranking versions of LR provide an additional 3-6% boost on the training MAP and 1% boost on the testing MAP compared with LR, which demonstrated the benefits of ranking-based learning in multimedia retrieval. The less significant improvement in the search set is partially due to the insufficiency of the training data for a single query. Since the difference between RLR and LR is not statistically significant, further experiments might be needed to verify the performance improvement of the proposed methods on other information retrieval

Data	Algorithms	TrainAP	TestAP	Prec10	Prec30	Prec100
t05	F-RLR	0.453	0.217	0.535	0.451	0.341
	RLR	0.447	0.217	0.529	0.433	0.341
	LR	0.389	0.207	0.506	0.433	0.341
t04	F-RLR	0.292	0.143	0.269	0.262	0.192
	RLR	0.283	0.141	0.269	0.264	0.192
	LR	0.261	0.132	0.238	0.241	0.184
t03	F-RLR	0.379	0.189	0.433	0.360	0.221
	RLR	0.371	0.186	0.433	0.342	0.224
	LR	0.358	0.185	0.431	0.358	0.229

Table 4.2: Comparison of rank learning algorithms on TRECVID'03-'05 data. TrainAP is the mean average precision on the development set. TestAP is the mean average precision on the search set. Prec10, Prec30 and Prec100 indicate the precision at the top 10, 30 and 100 retrieved shots.

tasks. Finally, we also observe that RLR, as an efficient approximation of its fully optimization version FRLR, achieved a fairly close performance to FRLR. Their differences on MAP are always less than 1% on three collections, which demonstrates RLR is a reasonable approximation for its fully optimization counterpart with a ten-fold speedup in the learning process.

4.5.3 Upper Bounds for Combination

In order to examine if query-independent combination is sufficient to support multimedia retrieval, we plot Figure 4.4 to compare three retrieval models including text retrieval baseline(Text), query-independent combination(LearnQI), query-specific combination(LearnQS), as well as two combination upper bounds, including the global linear bound(OracQI) and the local linear bound (OracRetr). Among these settings, LearnQI means learning uniform combination weights for all queries from the development data (where the upper bound is OracQI) and LearnQS means learning combination weights on a per query basis (where the upper bound is OracRetr). The upper bounds can be computed via the MCS algorithm by assuming the testing ground truth is known. From Figure 4.4, it is not difficult to find that although LearnQI could improve the retrieval perfor-

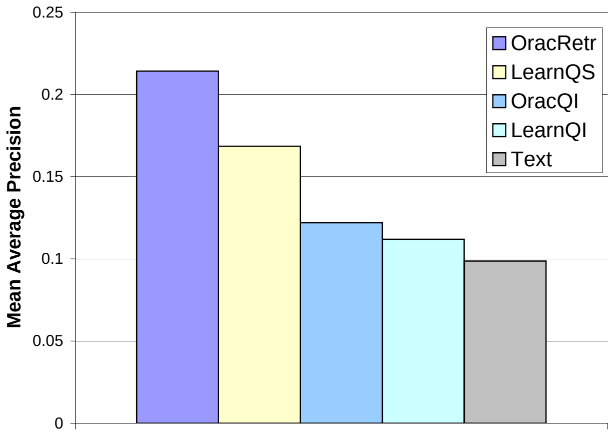


Figure 4.4: Comparing mean average precision of retrieval models and retrieval upper bounds averaged over TRECVID'02-'05.

mance over the text baseline, the mere 1% improvement is not so considerable as LearnQS (of which the improvement over baseline is statistically significant). This can be attributed to the fact that query-independent combination has much less flexibilities to capture the variation in the query space and it has limited potential to improve the retrieval outputs as reflected in the small difference between OracQI and Text. In contrast, using the query-specific learning strategy (LearnQS) can bring us more benefits by adaptively combining multiple knowledge sources. The local linear bound (OracRetr) even reveals the possibility to double the mean average precision of the text baseline if we can choose the optimal weights online. These observations provide strong evidence for us to investigate the usage of query-dependent combination approaches. However, since we cannot enumerate all the possible queries on the development data, learning weights on a per query basis is virtually impractical. Therefore in the next few sections, we will study several practical retrieval models that can handle query information in the combination process without enumerating every possible query.

4.6 Conclusion

This chapter presents a relevance-based probabilistic retrieval model to combine multiple knowledge sources in multimedia retrieval. Its connections to other retrieval models have been discussed. Based on this model, we provide a discriminative learning approach to estimate the combination weights from the training data. The experimental results on TRECVID'03-'05 have confirmed the superiority of using discriminative models to using generative models for parameter estimation, although the latter one is the most common choice in text retrieval.

In order to incorporate ranking information into the learning process, we develop a general margin-based rank learning framework for the information retrieval task. It aims to optimize the number of discordant pairs between the predicted ranking and the target ranking. We also propose an efficient approximation for the margin-based rank learning framework which can significantly reduce the computational complexity with a negligible loss in the performance. The experiments on three TRECVID collections demonstrate the effectiveness of the proposed rank learning algorithms. However, our further study on the limits of combination function uncovers that query-independent combination might not be sufficient to handle the complex multimedia combination task. Therefore, it might be desirable to investigate more advanced query-dependent combination approaches in future work.

4.7 Discussions

4.7.1 Weight Estimation

Generally the parameters of a classification problem can be estimated by two types of models: generative models and discriminative models [NJ02]. Discriminative models attempt to directly model the probability of class given the data, i.e., $P(c|x)$. But generative models alternatively estimates the class-conditional probability $P(x|c)$ as surrogates to find the maximal likely class based on Bayesian rules,

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

Traditionally, generative models have been a popular choice by estimating the class-conditional probabilities in relevance-based probabilistic models. The class-conditional probabilities include the probability of terms given relevance p_i and the probability of terms given irrelevance r_i. In the early years [RJ77], the parameters were mainly estimated from users' relevance feedback using a naive Bayes model. Later on, researchers directly estimated the parameters through explanatory data analysis by making simplifying assumptions on collections. For example, Croft [CH88] assumes p_i is a constant across the collection and q_i is the ratio between the number of documents containing i_{th} term and total number of documents. Therefore, the term weights can be translated into the widely used *idf* weights. The term-weight estimation can be further improved by making more accurate model assumptions. For instance, Van Rijsbergen [Rij79] proposed a simple tree-based model to capture the dependency between terms. A two-Poisson model extends the binary independence model under the assumption that p_i and r_i are both Poisson distributed. This work underlies the Okapi normalization algorithms.

The success of generative models largely depends on the validity of the model assumptions. However, these assumptions are not always true such as the independence assumption of term distributions. In contrast, a discriminative model(e.g. Logistic Regression, SVMs) typically makes less model assumptions and it prefers to "let data speak for its own". It has been preferred in many other domains of text processing such as text classification and information extraction. As pointed out by Vapnik [Vap95], "one should solve the (classification) problem directly and never solve a more general problem (class-conditional) as an intermediate step". There are some empirical results showing that discriminative models tend to have a lower asymptotic error as the training set size increased [NJ02]. Apart from theoretical considerations, we believe there are specific reasons for multimedia retrieval that make discriminative models a suitable choice. The following points explain some of them.

1) Various ranking features usually provide heterogeneous types of outputs which contains both query-dependent descriptors (retrieval experts) and query-independent descriptors (semantic concepts). Discriminative models can learn all the weights automatically in a unified way with less model assumption. Generative models might need to manually design different model distributions for ranking features, but some of the distributions might be difficult to define especially for the semantic concepts. Nallapati [Nal04] shows that with presence of heterogenous features, the discriminative models are superior to the generative models in a home-page finding task.

2) The dimensionality of the feature space of multimedia retrieval (an order of 100) is considerably smaller than that of text retrieval (an order of 10,000), which allows a discriminative model to estimate the parameters more robustly given the same relatively small amount of the training data provided.

4.7.2 Rank learning

The wide range of applications for rank learning has inspired numeric approaches to handle this problem especially in the context of information retrieval. One typical direction of rank learning is to formulate it into an ordinal regression problem, i.e., mapping the labels to an ordered set of numerical ranks. Herbrich et al. [HGO00] model the ranks as intervals on the real line, and optimize the loss function based on the true ranks and features. Following the similar idea, "PRank" [CS01] is developed based on an online linear learning algorithm called perceptron that uses one example at a time to update the linear feature weights. The ordinal regression formulation has been proven to be effective in the task of collaborative filtering. However, it might not be suitable for retrieval because the absolute rankings over documents are usually expensive to collect and users are less willing to provide such a detailed feedback in practice. Moreover, all the objects in ordinal regression have to be ranked in the same scale. But for retrieval, the ranking relationships only need to be consistent within a query which can greatly reduce the number of constraints.

As an alternative of learning the absolute numerical ranks, the approaches that model the relative ranking preferences between pairs of training data has also been investigated recently. In the setting of collaborative filtering, Freund et al. [FISS98] proposed the RankBoost algorithm which learns to rank a set of objects by combining multiple "weak" classifiers to build up a more accurate composite classifiers. The ranking SVMs proposed by Joachims [Joa02] is constructed on a risk-minimization framework with the goal to minimize the number of misorderings between the predicted ranks and target ranks. Bearing resemblance to the common classification SVMs, ranking SVMs can be solved with similar optimization techniques. Based on a simple probabilistic cost function, Burges et al. [Bea05] investigated a gradient descent method called RankNet to learn ranking functions with a neural network implementation. More recently, Chua et al. [CNG$^+$05] developed a ranking maximal figure-of-merit(MFoM) algorithm by maximizing the area under the ROC curve. This approach has been successful in the domain of video semantic feature extraction. In essence, the aforementioned rank learning algorithms transform ranks into a set of pairwise relationships between relevant and irrelevant examples and thus casts it into a classification problem built on example pairs. However, these algorithms usually suffer from an expensive training process due to the explosive amount of training data after coupling each pair of relevant and irrelevant documents, especially when the number of underlying training documents is large. For example, a query with 100 relevant documents and 900 irrelevant ones will result in 90,000 pairs of training examples, which is computationally intensive for many learning algorithms. It would be helpful to develop an efficient rank learning algorithm that is able to capture the ranking relationship while with a less learning time.

Chapter 5

Query Analysis

The aforementioned discriminative retrieval framework and its ranking counterpart provides a principled platform to support the task of retrieval source combination. However, in its current form, the combination parameters λ_i are still completely independent to the queries. In other words, the model will associate every possible query with the same set of combination parameters no matter what types of information needs users are expressing. In the previous chapter, we already showed that adopting such a query-independent knowledge combination strategy is not flexible enough to handle the variations of heterogeneous information needs.

Therefore, it is desirable to leverage the information from query description by developing more advanced retrieval methods. However, given the virtually infinite number of queries, it is impractical to learn combination functions on a per query basis. A trade-off needs to be found between the difficulty of providing training data and the ability of capturing the idiosyncrasy of each query. Following this argument, we propose a series of query analysis approaches which attempt to discover the mixing structure of past retrieval results and use the current query description as evidence to infer a better combination function.

5.1 Query-class Based Retrieval Model

In this section, we propose a retrieval approach called query-class based retrieval model, which aims to associate combination weights with a few pre-defined query classes and uses these weights to combine multi-modal retrieval outputs. In more detail, a user query is first classified into one of the four predefined query classes, i.e. finding named persons, named objects, general objects and scenes. Once

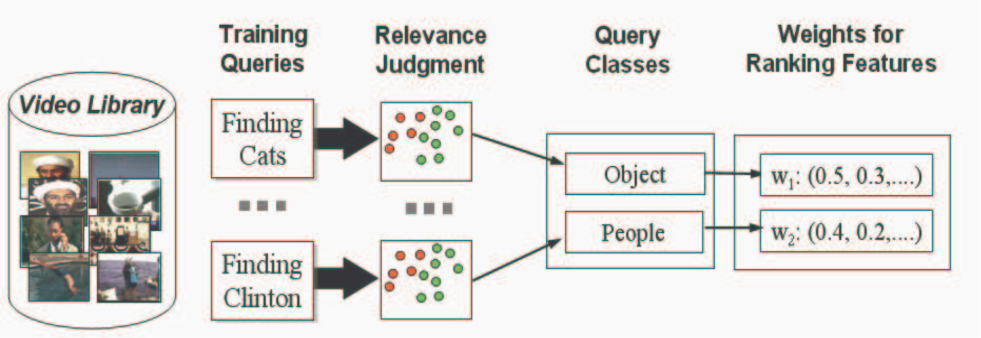


Figure 5.1: Illustration of query-class based retrieval model.

the query is categorized, the ranking features from multiple modalities can be combined with query-class associated weights. In this case, it is legitimate to collect truth data for each query class because the number of classes is very limited, while the learned weights can be reused for other unseen queries as long as they belong to some of the predefined classes. Figure 5.1 illustrates the process of query-class based retrieval model. The effectiveness of this model has been confirmed by our experiments on multimedia retrieval and many subsequent studies [CNL$^+$04, CNG$^+$05, Huu05, YXW$^+$05, KNC05].

5.1.1 Defining Query Classes

The design of query classes should follow two guidelines. First, the queries in the same query class should share similar combination functions. Second, queries should be automatically classified into one of the query classes with reasonable accuracy. After investigating the general queries for multimedia retrieval, we define the following four query classes according to the expressed intent[1]:

Named person (P-query) queries for finding a named person, possibly with certain actions, e.g., "Find shots of Yasser Arafat" and "Find shots of Ronald Reagan speaking".

Named object (E-query) queries for a specific object with a unique name, which distinguishes this object from other objects of the same type. For example,

[1] Our definition of query classes is different from the definitions of question categories in the TREC question answering track, such as searching for location, numeric number and description[LR02], because our purpose is to improve the fusion of multiple retrieval results instead of extracting exact answers from text archives.

"Find shots of the Statue of Liberty" and "Find shots of Mercedes logo" are such queries.

General object (O-query) queries for a certain type of objects, such as "Find shots of snow-covered mountain" and "Find shots of one or more cats". They refer to a general category of objects instead of a specific one among them, though they may be qualified by adjectives or other words.

Scene (S-query) queries depicting a scene with multiple types of objects in certain spatial relationships, e.g., "Find shots of roads with lots of vehicles" and "Find shots of people spending leisure time on the beach". They differ from O-queries mainly by the number of the object types involved.

According to our definition, each query class should favor a specific set of ranking features. For example, face presence, size, position information and face recognition are critical to P-queries but of little value to other query classes. For both P-queries and E-queries, the text transcript is particularly important since such queries are more likely to have a perfect match in textual features, and so is video OCR because proper names may appear on the screen as overlaid text. On the other hand, visual features like color, texture, and shape can be helpful to the O-query and S-query. Overall, such a query classification approach captures the characteristics of queries regarding the feature effectiveness and therefore is promising for leading to a better performance.

Since the above query classes are mainly designed for news video collections, they may not be perfect for other arbitrary video collections. For example, movie archives may not have as many P-queries as news video archives, but they may instead have more S-queries. But the idea of query-class based retrieval is generally applicable. Once the underlying contents are changed, we only need to re-design the query classes without changing the entire framework.

5.1.2 Query Classification by Text Processing

In the following discussions, we mainly consider dealing with TRECVID queries as a standard set of queries with rich text descriptions, but the proposed method is also applicable for queries in other domains. The queries from TRECVID are generally in a regular form and primarily factual queries, but the size of existing query pool is relatively small so far. Therefore, query classifiers based on data-driven statistical inference might not be a good choice. Alternatively, we resort to rule-based text processing techniques to classify the queries. They consist of three phases: named entity extraction, tagging and chunking, and syntactic parsing.

Named entity extraction is mainly used to identify queries that contain proper names like people, location, or organization. Our named entity extraction method is similar to that used by MITRE [MMM97] and BBN [BMSW97]. First, every word in a corpus of broadcast news transcript is manually labeled with a named-entity tag like *-person-*, *-location-*, *-organization-*, *-time-*, and *-none-*, resulting in a sequence alternating between words and tags, such as "*-person-* Barry, *-person-* Serafin, *-organization-* ABC, *-organization-* news, *-none-* in, *-location-* Washington". A trigram model is then learned from this tag-word sequence using a statistical language modeling toolkit [CR97]. To detect named entities from unlabeled text, we build a Hidden Markov Model (HMM) that models each type of named-entity tag as a state and words as the observations of each state. The transition and emission probabilities of the HMM are provided by the trigram model. The Viterbi algorithm is used to find the most likely sequence of states (named-entity tags) that generates the observed word sequence in the given text. After extracting the named entities, we classify those queries containing people names as P queries and those containing organization and/or location names as E queries. Queries with both people and organization/location names are also classified into the P class.

After P-queries and E-queries are recognized in the previous stage, some preliminary text analysis including part-of-speech(POS)-tagging and NP-chunking is conducted on the remaining queries to distinguish O-queries and S-queries. This starts with applying Brill's transformation-based POS tagger [Bri94] on each query to obtain the POS of each query word. The tagged query is then fed into a text chunker, namely the baseNP chunker [RM95], which can divide the query into segments such as noun phrases (NP) and verb phrases (VP). Because one major distinction between queries for general objects and scenes is the number of (different) objects, we label the queries with one longest-matched NP as O-queries, and those with multiple longest-matched NPs as S-queries. It is intuitive to count the number of objects by the number of NPs in the query, as objects are normally referred by noun phrases. We only count the longest-matched NPs because NPs can be nested. For example, the NP "a pink flower" recursively contains another NP "flower", both of which refer to the same object. Thus, the number of the longest NPs better approximates the number of objects a user intends to specify.

Although the method described above works reasonably well in general, it has an obvious drawback because a single longest NP may refer to multiple objects. For instance, our method might classify the S-query "a person diving into water" and "balloon in the sky", as O-queries by mistake. Therefore, syntactic parsing is applied to correct those misclassified S-queries. Specifically, we send all the queries with one longest NP to Link Grammar Parser[GLS95] so as to produce the syntactic structure of the queries. In this way, the internal structure of the longest NP can be analyzed. For example, "a person diving into water" is parsed as *NP (NP (a person) VP (diving PP (into (NP water))))*, where PP denotes prepositional phrase. The queries with a single NP which actually contains multiple NPs inside are re-assigned to S-queries, while the rest remain as O-queries. The only exception is queries with *NP PP (prep NP)* structure but having "of" as its preposition, such as "a cup of coffee", which actually refers to only one object since the NP after "of" is normally used to modify the NP before it. Hence, queries with such structure are classified as O-queries. This query classification method based on superficial text analysis is able to classify 93% TRECVID queries correctly as shown in our experiments. Note that this work does not necessarily depend on the rich description of the topic. Our query classification technique is also applicable when users only provide several simple keywords by using the named entities from transcripts.

5.1.3 Parameter Estimation

After each query is associated with a single query class, the parameters can be estimated similarly as Eqn(4.11) except the training data are restricted in the given query class. In more detail, the maximum-likelihood estimation for the parameter λ can be written as,

$$\begin{aligned} \lambda_k^* &= \arg\max_{\lambda} \prod_{t=1}^{M_Q} \prod_{j=1}^{M_D} P(y_+|D_j, Q_t)^{y'_{tj}} P(y_-|D_j, Q_t)^{1-y'_{tj}} \\ &= \arg\max_{\lambda} \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} \log \left[\sum_k \delta(z_{k+}|Q_t) \cdot \sigma \left(y_{tj} \sum_{i=1}^{N} \lambda_{ki} f_i(D_j, Q_t) \right) \right], \end{aligned}$$

where $P(y_+|D_j, Q_t)$ is the probability of relevance defined in Eqn(4.5) and $\delta(z_{k+}|Q_m)$ is the delta function which is set to 1 when Q_m belongs to k^{th} query class and 0 otherwise. The optimization of the log-likelihood function can be achieved similarly as optimizing a logistic regression problem.

5.2 Experiments

In this section, we present some experimental results for the proposed query-class based retrieval models. We first describe a re-ranking retrieval model designed for news video collections and then evaluate the performance for both automatic query classification and video retrieval on the TRECVID'02-'05 collections to demonstrate the usefulness of query analysis.

5.2.1 A Reranking Model for News Video Retrieval

Although multimedia retrieval can be analogous to the meta-search problem, it bears distinct characteristics. This section presents two observations specific to video retrieval from news archives, and then details a re-ranking framework to merge results from multiple knowledge sources.

Based on evidence from the best-performing video retrieval systems in previous NIST TREC Video Retrieval evaluation tasks, text features are the most reliable ranking features for selecting semantically relevant documents. However, ranking of video shots cannot simply rely on these text features. One important reason is that words may be spoken in the transcript when no associated images are present, e.g. a news anchor might discuss a topic for which no video clips are available. A reporter may also speak about a topic with the relevant footage following later in the story, resulting in a major time offset between the word and the relevant video clips. As shown in Figure 5.2(a), text retrieval will at times prefer the shots of studio settings or anchors, however this is usually not desirable. Moreover, word sense ambiguity may result in videos retrieved of the wrong meanings, e.g. either a river shore or a financial institution is possible for the word 'bank'. Speech recognition errors or VOCR errors may also result in incorrect retrieval. In general, retrieval using text features exhibits satisfactory recall but relatively low precision [HBC$^+$03].

Fortunately, other knowledge sources from various video modalities can be used to rerank the text retrieval output. These sources include audio features, motion vectors, visual features (e.g. color, edge and texture histograms) and any number of high-level semantic concepts. Generally speaking, none of these alone can capture the full content of the multimedia documents and therefore retrieval results based only on these features are mostly unsatisfying. Figure 5.2(b) depicts the top results of image retrieval that returns nothing related to the query. However, these weak features can provide some indication of how closely the video documents are related to the specific query descriptions. They can also be

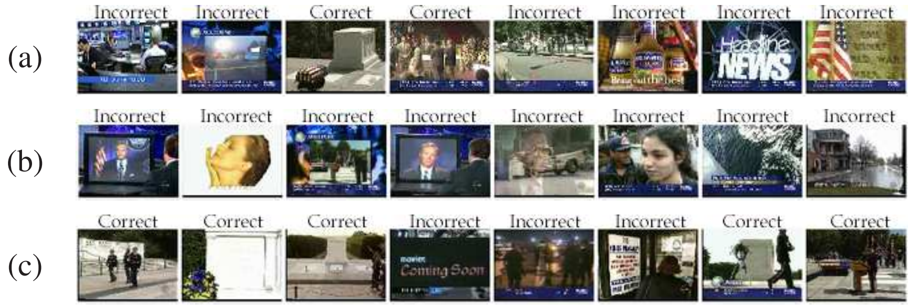

(a)
Incorrect Incorrect Correct Correct Incorrect Incorrect Incorrect Incorrect
(b)
Incorrect Incorrect Incorrect Incorrect Incorrect Incorrect Incorrect Incorrect
(c)
Correct Correct Correct Incorrect Incorrect Incorrect Correct Correct

Figure 5.2: The key frames of top 8 retrieved shots for query "Find the Tomb of Unknown Soldiers at Arlington National Cemetery" using (a) text features (b) image features (c) text plus image features with anchors and commercials filtered out.

used to filter out irrelevant documents such as anchor persons or commercials. Figure 5.2(c) illustrates the advantage of weak features which can augment text retrieval by finding the similar objects and filtering news anchor together with commercial shots.

These observations indicate that for news video retrieval, we should rely on text-based retrieval as the major source to answer semantic queries, while using the weak features from other modalities in combination to refine the ranking from text retrieval. Therefore, we propose the following re-ranking framework to generate the retrieval results,

$$\log O(y|D,Q) = \begin{cases} \sum_{i=1}^{N} \lambda_i f_i(D,Q), & D \in \mathcal{D}_R; \\ \sum_{i=1}^{N} \lambda_i f_i(D,Q) - C_0, & D \notin \mathcal{D}_R. \end{cases} \tag{5.1}$$

where λ_i is the weight for the t^{th} descriptor, $\mathcal{D}_R$ is the set of relevant video shots generated by text retrieval. We choose $\mathcal{D}_R$ to be the top retrieved documents from text retrieval. C_0 is a large constant to ensure the other weaker features could only re-rank the documents provided inside $\mathcal{D}_R$. Note that, the choice of C_0 is not important as long as it can move all the text-relevant documents on top of the other non-relevant documents. Then the other weaker features can be used to re-rank the outputs just on these top-ranked shots.

In the learning phase, because the goal of retrieval is to improve the performance in terms of average precision or precision at top-ranked shots, the decision loss is rapidly discounted as the ranks of documents become closer to the bottom. Therefore we only need to consider the top-ranked training data for each query, i.e, in this case the top-ranked shots found by text retrieval. Therefore the re-ranking

Query Class		**P**	**E**	**O**	**S**	**Overall**
Truth Labels		14	12	30	28	84
QC1	Correct	14	10	28	21	73(87%)
	Miss	0	2	2	7	11(13%)
	False Alarm	1	1	8	1	11(13%)
QC2	Correct	14	10	28	26	78(93%)
	Miss	0	2	2	2	6(7%)
	False Alarm	1	1	3	1	6(7%)

Table 5.1: Effectiveness of query classification.

model can enjoy the advantages of learning with a considerably smaller training set and more balanced training data with respect to the ratio between positive and negative data.

5.2.2 Query Classification Results

The proposed query classification method is evaluated using a total of 84 TREC'02 - '04 queries as the testbed. Each testing query is manually labeled with the query class it belongs to. All the manual labels are checked by two human subjects without any disagreements found. Then the manual labels are compared with the class labels assigned by our query classification method. The comparison results are summarized in Table 5.1, where QC-1 and QC-2 refer to the methods without and with syntactic parsing respectively.

Our method works reasonably well and quite reliably in classifying the TREC queries, achieving 87% without using syntactic parsing and 93% with parsing. P-queries and E-queries are almost perfectly classified, which implies the effectiveness of our named entity extraction algorithm. For QC-1, the large number of false alarms on O-query and misses on S-query are due to its mistaking some S-queries with only one longest-matched NP as O-queries. We can see that QC-2 by syntactic parsing corrects 5 out of 8 such misclassifications. It fails to correct the other 3 due to the parsing errors caused by complicated structures of the NPs. For example, when parsing "a hot air balloon in the sky", our parser erroneously puts "air" as a noun and "balloon" as a verb. Besides parsing errors, another source of errors is the implicit priority of query classes. For example, the only false alarm on a P-query is because an E-query "Find shots of the Price Tower designed by Frank

Data Set	t02o	t03o	t04o	t05o	t03e	t04e	t05e
Query Num	25	25	24	24	25	24	24
Doc Num	24263	75850	48818	77979	75850	48818	77979

Data Set	-	t03d	t04d	t05d	t04dy	t04dz	t04dx
Query Num	-	25	24	24	40	64	88
Doc Num	-	47531	124097	74532	124097	124097	198629

Table 5.2: Labels of video collections and their statistics. $t * *d$ indicate development sets and $t04dx/y/z$ are the external development sets. $t * *o/e$ indicate search sets with original query keywords or expanded query keywords.

Lloyd Wright" has been classified as a P-query, since higher priority is given to the P-query over the other named entities. We also realize that the query pool is still not large enough to provide convincing results for other types of queries. But our results show that automatic query classification has achieved reasonably high accuracy to support the following video retrieval process.

5.2.3 Retrieval Results

Our experiments are designed based on the guidelines of the manual retrieval task in the TREC video retrieval evaluation(TRECVID), which requires an automatic video retrieval system to search relevant documents without any human feedback. In this task, the retrieval units are video shots defined by a common shot boundary reference. The proposed combination algorithms are evaluated using the queries and the video collections officially provided by TREC '02-'05 [SO03]. For TREC'03-'05, each of these video collections is split into a development set and a search set chronologically by source. The development sets are used as the training pool to develop automatic multimedia retrieval algorithms and the search sets mainly serve as the testbeds for evaluating the performances of retrieval systems. For each query topic, the relevance judgment on search sets was provided officially by NIST. The relevance judgment on development sets was collaboratively collected by several human annotators using the Informedia client [HCC$^+$04]. Each search set from TREC'03-'05 is coupled with two types of text queries, i.e., original keywords extracted from official descriptions and expanded keywords added by human annotators (named t**o/e respectively). Moreover, we manually designed 40 additional queries and collected the ground truth on the TREC'04 development set in order to demonstrate that the query-class based retrieval model is able to learn from arbitrary queries that do not belong to the test-

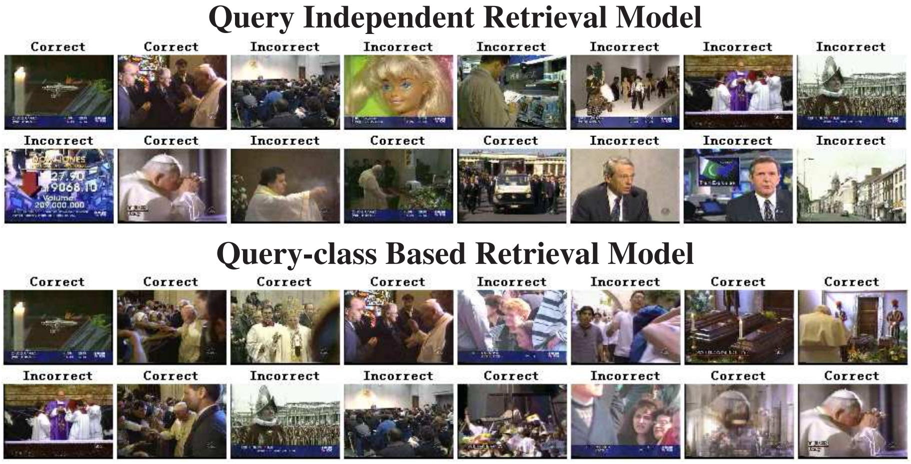


Figure 5.3: The keyframes of the top 16 retrieved video shots for the query "Finding shot of Pope John Paul II". The words "corret/incorrect" above indicate whether the corresponding shots are relevant to the query or not.

ing set. This training set is called $t04dy$. We merge it with the TREC'04 queries and with both TREC'04-'05 queries to construct two external training corpora called $t04dz$ and $t04dx$. Table 5.2 lists the labels of video collections and their query/document statistics.

As the building blocks for multimedia retrieval, we generated a number of ranking features on each video document, including 14 high-level semantic features learned from development data (*face, anchor, commercial, studio, graphics, weather, sports, outdoor, person, crowd, road, car, building, motion*), and 5 uni-modal retrieval experts (*text retrieval, face recognition, image-based retrieval based on color, texture and edge histograms*). The detailed description on the feature generation process can be found in [HCC$^+$04]. To avoid the problems brought by inconsistent scales of various retrieval outputs, we transformed the raw scores into their ranks in each ranking feature and normalized them into the range of [0,1] where the highest ranked document corresponds to one and the lowest ranked document is zero. Moreover, in an attempt to incorporate the ranking information in the learning process, we weighted the positive data stronger to balance the positive/negative data distribution and meanwhile shifted the median value of each feature to zero [YH06a].

Figure 5.3 illustrates the advantage of using the query-class based retrieval model over the query independent retrieval model. There are 10 relevant shots

Data	**Method**	**MAP**	**P30**	**P100**	**Person**	**SObj**	**GObj**	**Sport**	**Other**
t02o	Text:N/A	0.098(+0%)	0.113	0.076	0.124	0.183	0.079	0.023	0.011
	QI:t04dy	0.123(+25%)	0.123	0.089	0.261	0.212	0.067	0.007	0.023
	QI:t04dz	0.123(+25%)	0.123	0.089	0.261	0.212	0.067	0.007	0.023
	QI:t04dx	0.114(+16%)	0.135	0.081	0.199	0.205	0.074	0.007	0.019
	QC:t04dy	0.125(+27%)	0.115	0.080	0.288	0.205	0.076	0.002	0.009
	QC:t04dz	0.132(+34%)	0.119	0.082	0.315	0.206	0.076	0.002	0.009
	QC:t04dx	**0.132(+34%)**	0.131	0.086	0.316	0.207	0.075	0.004	0.013
t03o	Text:N/A	0.146(+0%)	0.171	0.118	0.371	0.230	0.068	0.031	0.007
	QI:t04dy	0.156(+7%)	0.207	0.134	0.272	0.331	0.081	0.096	0.011
	QI:t04dz	0.146(+0%)	0.204	0.132	0.269	0.286	0.079	0.096	0.010
	QI:t04dx	0.150(+2%)	0.212	0.136	0.251	0.293	0.095	0.101	0.012
	QC:t04dy	0.188(+29%)	0.227	0.139	0.403	0.336	0.087	0.126	0.011
	QC:t04dz	0.199(+37%)	0.239	0.140	0.397	0.338	0.086	0.127	0.011
	QC:t04dx	**0.200(+37%)**	0.236	0.137	0.407	0.336	0.088	0.106	0.015
t03e	Text:N/A	0.192(+0%)	0.231	0.155	0.402	0.296	0.103	0.134	0.044
	QI:t04dy	0.184(-4%)	0.251	0.175	0.284	0.345	0.109	0.225	0.027
	QI:t04dz	0.177(-8%)	0.248	0.174	0.279	0.297	0.112	0.261	0.024
	QI:t04dx	0.184(-4%)	0.243	0.180	0.260	0.311	0.128	0.289	0.027
	QC:t04dy	0.227(+18%)	0.283	0.178	0.438	0.348	0.126	0.318	0.020
	QC:t04dz	0.236(+23%)	0.289	0.180	0.405	0.346	0.125	0.321	0.020
	QC:t04dx	**0.239(+24%)**	0.289	0.187	0.409	0.345	0.127	0.293	0.036

Table 5.3: Comparison of combination approaches, including text retrieval, query-independent combination and query-class based combination(I).

retrieved out of top 16 shots with query-class based retrieval, compared with 5 relevant shots retrieved with the query-independent retrieval. This statistically significant improvement is achieved by a successful combination of different knowledge sources, such as the face detector and the image retrieval expert, using the parameters learned for the person-finding query.

Table 5.3 and Table 5.4 list a more detailed comparison for various retrieval approaches in terms of mean average precision at top 400 shots and precision/recall at 30 and 100 shots. We compare four baseline combination approaches , i.e., text retrieval(**Text**) and query-independent combination on three training collections(**QInd:t04dx/y/z**), with three configurations of the query-class based combination model(**QClass:t04dx/y/z**). From the results, we can find that the improvement of QInd over Text is not as stable as that of QClass in all the collections. For instance, in the collection of t03o and t03e, learning the query-independent combination weights produces even worse results than the text baseline, which indicates the difficulty of knowledge combination in multimedia retrieval. In contrast, QClass is almost always superior to QInd in terms of both mean average

Data	Method	MAP	P30	P100	Person	SObj	GObj	Sport	Other
t04o	Text:N/A	0.078(+0%)	0.178	0.107	0.188	0.012	0.033	0.046	0.044
	QI:t04dy	0.088(+12%)	0.172	0.109	0.194	0.047	0.030	0.095	0.049
	QI:t04dz	0.084(+7%)	0.177	0.112	0.192	0.025	0.032	0.088	0.041
	QI:t04dx	0.079(+0%)	0.177	0.116	0.144	0.063	0.034	0.108	0.051
	QC:t04dy	0.088(+12%)	0.157	0.101	0.215	0.080	0.046	0.087	0.017
	QC:t04dz	0.084(+7%)	0.158	0.103	0.198	0.082	0.046	0.089	0.017
	QC:t04dx	**0.094(+20%)**	0.199	0.125	0.194	0.080	0.046	0.108	0.045
t04e	Text:N/A	0.097(+0%)	0.184	0.125	0.199	0.003	0.041	0.152	0.048
	QI:t04dy	0.112(+15%)	0.203	0.136	0.206	0.004	0.037	0.215	0.066
	QI:t04dz	0.110(+13%)	0.209	0.140	0.204	0.004	0.037	0.219	0.061
	QI:t04dx	0.103(+5%)	0.191	0.144	0.157	0.006	0.036	0.232	0.072
	QC:t04dy	0.107(+9%)	0.187	0.132	0.227	0.005	0.045	0.197	0.032
	QC:t04dz	0.103(+5%)	0.187	0.133	0.211	0.006	0.045	0.200	0.032
	QC:t04dx	**0.113(+16%)**	0.212	0.152	0.206	0.005	0.046	0.233	0.056
t05o	Text:N/A	0.073(+0%)	0.207	0.175	0.141	0.015	0.097	0.075	0.016
	QI:t04dy	0.103(+41%)	0.272	0.198	0.194	0.026	0.105	0.170	0.017
	QI:t04dz	0.108(+47%)	0.269	0.203	0.189	0.025	0.096	0.232	0.016
	QI:t04dx	0.105(+44%)	0.268	0.205	0.164	0.029	0.090	0.271	0.017
	QC:t04dy	0.113(+54%)	0.268	0.191	0.162	0.032	0.101	0.326	0.014
	QC:t04dz	0.116(+59%)	0.272	0.195	0.173	0.032	0.101	0.327	0.014
	QC:t04dx	**0.116(+58%)**	0.292	0.211	0.173	0.031	0.100	0.322	0.017
t05e	Text:N/A	0.103(+0%)	0.253	0.210	0.178	0.038	0.095	0.181	0.029
	QI:t04dy	0.139(+34%)	0.347	0.243	0.230	0.066	0.103	0.291	0.034
	QI:t04dz	0.141(+36%)	0.347	0.245	0.227	0.064	0.095	0.335	0.033
	QI:t04dx	0.146(+40%)	0.354	0.251	0.218	0.070	0.086	0.395	0.032
	QC:t04dy	0.144(+38%)	0.317	0.241	0.201	0.081	0.097	0.418	0.022
	QC:t04dz	0.160(+54%)	0.343	0.248	0.255	0.080	0.097	0.422	0.022
	QC:t04dx	**0.168(+62%)**	0.376	0.265	0.258	0.079	0.096	0.452	0.033

Table 5.4: Comparison of combination approaches, including text retrieval alone, query-independent combination and query-class based combination(II).

precision and precision at several levels. The margin between QClass and QInd is quite significant for some collections such as t03e/o and t05e/o. Generally speaking, the more training queries we use to estimate the combination parameters, the higher performance QClass can achieve. This can be verified by the fact that learning from $t04dx$ can usually bring the highest average precision among all three training settings in QClass. On average, QClass provides a roughly 35% improvement over the text retrieval baseline and a 17% improvement over QInd in terms of MAP [2]. To analyze the results in more detail, we also grouped the queries

[2] The highest MAP for QClass is also higher than the corresponding performance of the best

Data	**t03d**	**t04d**	**t05d**	**t04dy**	**t04dz**	**t04dx**
t02o	0.102	0.136	0.091	0.125	0.132	0.132
t03o	0.173	0.187	0.172	0.188	0.199	0.200
t03e	0.212	0.221	0.208	0.227	0.236	0.239
t04o	0.089	0.087	0.083	0.088	0.084	0.094
t04e	0.109	0.111	0.098	0.107	0.103	0.113
t05o	0.099	0.110	0.108	0.113	0.116	0.116
t05e	0.129	0.162	0.147	0.144	0.160	0.168

Table 5.5: Query class combination with different training sets.

in each collection and reported their MAPs in five different categories, i.e., named person (**Person**), special object(**SObj**), general object (**GObj**), sport (**Sport**) and general (**Other**) queries. By comparing MAPs with respect to each query category, we find that QClass benefits most from the **Person, Sport and SObj** type queries. However, it does not produce better results for the **GObj** and **Other** type queries, which might indicate these type of queries need to be further refined into several smaller categories.

To further investigate the sensitivity of the query-class based combination models with respect to the choice of training and search collections, we also list Table 5.5 which shows the QClass retrieval performance when the combination parameters are learned from six development collections. Surprisingly, the retrieval results do not benefit from estimating the combination parameters from the development data in the same year. For example, learning from $t03d$ does not provide the best average precision in $t03o/e$. On the contrary, the parameters learned from larger training collections (such as $t04dx$) are usually of better quality than the parameters learned from smaller training collections (such as $t03d$). This observation demonstrates that a good query-class combination model is universally effective for different search collections without being too sensitive to the corpus variation, whereas collecting sufficient training data is a necessary condition to construct models with high quality.

TREC submission in each year, according to the results found found in the TRECVID website [SO03].

5.3 Probabilistic Latent Query Analysis(pLQA)

Despite recent successes, query-class combination methods still have plenty of room for improvement. One major issue is that query classes usually need to be defined using expert domain knowledge. This manual design can work well when only a few query classes are needed, but it will become difficult for tens or hundreds of query classes, where each query in a class has to share similar characteristics and thus a similar combination strategy. If the query classes are not carefully defined, a large learning error from both the query-class assignment and the combination parameter estimation might result. Furthermore, current query-class methods do not allow mixtures of query classes, but at times such a mixture treatment could be helpful. For instance, the query "finding Bill Clinton in front of US flags" should be associated with both a "Person" class and a "Named Object" class rather than only one of these. Finally, determining the number of query classes remains to be an unanswered problem in these methods due to the nature of manual design. Some previous approaches [VGJL95, KNC05] can discover query classes by using clustering techniques. However, these approaches typically separate the processes of query class categorization and combination optimization into two sequential steps without being jointly optimized. Moreover, it is not straightforward for general clustering methods to handle mixtures of query classes in a principled manner.

Based on these considerations, it is desirable to develop a data-driven probabilistic combination approach that allows query classes and their corresponding combination parameters to be automatically discovered from the training data itself, rather than handcrafted using human knowledge. Therefore, we propose a new combination approach called probabilistic latent query analysis (pLQA) to merge multiple retrieval sources based on statistical latent-class models. The proposed approaches have advantages over query-independent and query-class combination methods in several ways: (1) they unify combination weight optimization and query-class categorization into a single discriminative learning framework; (2) they are able to automatically discover the latent query classes directly from training data; (3) they can handle mixture of query classes in one query and (4) they can determine the number of query classes with an statistical model selection principle. Experiments are conducted on two retrieval applications, i.e., multimedia retrieval on the TRECVID'02-'05 collections [SO03] and meta-search on the TREC-8 collection [VH99]. The results show that the proposed approaches can uncover sensible latent classes from training data, and also demonstrate higher effectiveness in combining multiple retrieval sources. In the rest of this section, we first discuss the basic form of the pLQA method and its parameter estimation strategies. To deal with unseen queries outside the training collection, we then extend pLQA to its adaptive version and kernel version.

5.3.1 Basic pLQA

It would be ideal if we could learn specific combination parameters for every possible query. However, given the virtually infinite number of query topics, it is impractical to learn the combination weights on a per query basis because we cannot collect enough training data individually. We need to come up with a trade-off to balance the difficulties of collecting training data and the ability to capture the idiosyncracy of the query space. To achieve this, we make the following assumptions in our models: (1) the entire query space can be described by a finite number of query classes, where queries from each class share the same combination function; (2) the query description can be used to indicate which class a query belongs to. Under the first assumption, the basic probabilistic retrieval model presented in the last chapter can be naturally extended to a finite mixture of conditional probabilistic models. Formally, we can introduce a multinomial latent query class variable z to indicate which mixture the combination function is drawn from. Based on the second assumption, the choice of z is solely depending on the query Q. Putting all these together, we have the joint probability of relevance y and latent variable z as,

$$P(y_+, z|Q, D; \mu, \lambda) \quad = \quad P(z|Q; \mu)P(y_+|Q, D, z; \lambda), \tag{5.2}$$

where μ is the parameter for multinomial distributions, λ is the combination parameter for query classes. The mixture component $P(y_+|Q, D, z; \lambda)$ corresponds to a single logistic regression model and the mixing proportion $P(z|Q; \mu)$ controls the switches among different classes based on the query-dependent parameters μ_Q. By marginalizing out the hidden variables z, the corresponding mixture model can be written as, (M_z is the number of query classes)

$$P(y_+|Q, D; \mu, \lambda) = \sum_{z=1}^{M_z} P(z|Q; \mu) \cdot \sigma \left(\sum_{i=1}^{N} \lambda_{zi} f_i(D, Q) \right). \tag{5.3}$$

In the following discussions, we refer the model presented in Eqn(5.3) to as the *basic pLQA*(**BpLQA**) model where each latent query class represents a group of similar queries sharing the same combination weights. Note that when the

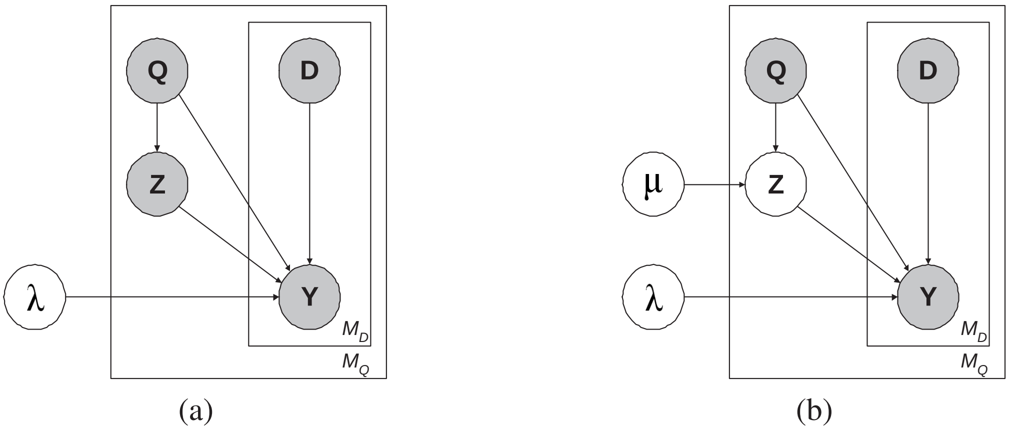


Figure 5.4: Graphical model representation for (a) query-class combination models where the query classes are manually defined, (b) probabilistic latent query analysis(pLQA) where the query classes are defined as latent variables. The nodes with known values are shaded, while other nodes are unshaded. The plates stand for replicas of the subgraphs where the number of replicas is on the corner.

number of latent variables is reduced to 1, BpLQA degrades to the case where retrieval source combination is not relevant to queries, i.e., a query-independent combination approach.

Remark 1: Figure 5.4(ab) compare the probabilistic graphical model representations of BpLQA and the query-class combination methods. One of the their major differences is the semantic of the mixing proportions $P(z|Q, \mu)$. In the query-class method, query classes have to be derived from manually defined generation rules before the learning process. However, query classes in the BpLQA model are expressed as latent variables and thus can be estimated directly from training data. Moreover, they also differ in the way how they associate queries with query classes. By analogy, the query-class method can be viewed as a “hard” query categorization that associates each query with one single query class, and meanwhile BpLQA can be viewed as a “soft” query categorization that leads to a probabilistic membership assignment from queries to latent query classes.

Because of these representation differences, BpLQA can enjoy the following advantages over the query-class combination method: (1) it can automatically discover the query classes from training data rather than by manual definition, (2) it offers probabilistic semantics for the latent query classes and thus allows mixing multiple query classes for a single query, (3) it can discover the number of query classes in a principled way, (4) it can address the insufficient data problem caused by ill-defined query classes that have few positive examples in the training set and (5) it unifies the combination weight optimization and query class categorization into a single learning framework.

Remark 2: BpLQA is also related to the aspect model or *probabilistic latent semantic indexing*(pLSI) [Hof99] in document distribution modeling, which assume the words are generated from a set of latent aspects z independently from the document identity d. However, BpLQA differs from pLSI in the sense that it assumes the latent aspects are imposed on the query space instead of the document space. The number of query mixtures are usually smaller than the document mixtures due to the sparseness of training data. Moreover, instead of estimating the underlying word-document distribution, the main goal of BpLQA is to estimate the combination parameter λ and predict the relevance of documents for retrieval.

The construction of BpLQA naturally corresponds to the following sampling process for each relevance judgement Y given a pair of document D and query D,

1. Choose a query class $z \sim$ Mult(μ_Q) for query Q
2. Choose the relevance $y \sim$ Bernoulli(p), where $logit(p) = \sum_i \lambda_{ki} f_i(D, Q)$.

Remark 3: On the level of the entire collection, BpLQA models the joint probability by multiplying the conditional probability as follows,

$$P(\{y_{t+}\}|Q, \{D_t\}) = \prod_t \sum_z P(z|Q; \mu)\sigma\left(\sum_{i=1}^N \lambda_{zi} f_i(D_t, Q)\right).$$

It is instructive to compare the proposed model with another seemingly similar mixture model as follows,

$$P(\{y_{t+}\}|Q, \{D_t\}) = \sum_z P(z|Q; \mu) \prod_t \sigma\left(\sum_{i=1}^N \lambda_{zi} f_i(D_t, Q)\right).$$

Under this mixture model, each collection is first associated with one query class and then all of the document relevance is generated independently under the same query class. In some senses, it models the collections under the assumption that each query only exhibits one class. In contrast, BpLQA allows associating documents with different query classes even for a single query. As shown in previous work [BNJ03], this flexibility is very effective in modeling a large number of documents without any additional parameters introduced in the model.

5.3.2 Parameter Estimation

The parameters in BpLQA can be determined by maximizing the incomplete data log-likelihood function,

$$l(\mu, \lambda) = \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} \log \left[\sum_{z} \mu_{zt} \sigma \left(y_{tj} \sum_{i=1}^{N} \lambda_{zi} f_i(Q_t, D_j) \right) \right],$$

where $\mu_{zt} = P(z|Q_t; \mu)$ is the probability of choosing hidden query classes z given query q_m, λ_{zi} is the weights for $f_i(\cdot)$ under the class z. The parameters in BpLQA can be estimated by maximizing its incomplete data log-likelihood. A typical approach to achieve this is to use the Expectation-Maximization (EM) algorithm [DLR77], which can obtain a local optimum of log-likelihood by iterating E-step and M-step until convergence. The E-step of the BpLQA model can be derived by computing the expectation of z,

$$h_{tj}(z) = P(z|Q_t, D_j) = \frac{\mu_{zt}\sigma(y_{tj}\sum_i \lambda_{zi} f_i(Q_t, D_j))}{\sum_{z=1}^{M_z} \mu_{zt}\sigma(y_{tj}\sum_i \lambda_{zi} f_i(Q_t, D_j))},$$

where $\mu_{zt} = P(z|Q_t; \mu)$ is the probability of choosing hidden query classes z given query q_m and λ_{zi} is the weights for $f_i(\cdot)$ under the class z. By optimizing the auxiliary Q-function, we can derive the following M-step update rules,

$$\begin{aligned} \lambda_{zi} &= \arg\max_{\lambda_k} \sum_{jt} h_{tj}(z) \log \left[\sigma \left(\sum_{i=1}^{N} \lambda_{zi} f_i(Q_t, D_j) \right) \right], \\ \mu_{zt} &= \frac{1}{M_D} \sum_{j=1}^{M_D} P(z|Q_t, D_j). \end{aligned}$$

When the log-likelihood converges to a local optimum, the estimated parameters can be plugged back into the BpLQA model to describe the underlying query space structure and show how the training queries are organized.

The choice of the mixture number K depends on the amount and the distribution sparseness of the training data. As the number of latent variables grows, the family of decision functions represented in BpLQA will become less restricted and thus lead to a lower bias in the estimated models, but meanwhile it would suffer from a higher variance due to an increased model complexity. Based on the bias-variance trade-off, we can determine the number of query classes that is sufficient to capture the variations in the query space with the given amount of training data.

To be more concrete, the number of query classes can be obtained by maximizing the sum of log-likelihood and some model selection criteria such as Akaike Information Criteria(AIC), Bayesian Information Criteria(BIC) and so forth. In our work, we choose BIC [Sch78] as the selection criterion. It quantifies the relative goodness-of-fit of statistical models by maximizing the formula of

$$BIC = 2l(\mu, \lambda) - k \log(n),$$

where k is the number of parameters to optimize and n is the number of training data. The second term in BIC corresponds to a model-complexity regularization statistic. It has a solid probabilistic interpretation that the model with the highest BIC is the model with the highest posterior probability given the data under some general conditions.

5.3.3 Adaptive pLQA

Discovering the underlying structure of query space by itself is not sufficient to handle the retrieval source combination, because a practical combination model should be able to predict combination parameters for unseen queries outside the training collection. Unfortunately, BpLQA cannot easily generalize the multinomial parameters μ to any of these unseen queries, because each parameter $\mu_{\cdot t}$ in BpLQA specifically corresponds to the t^{th} training query. Since it is impossible to enumerate all possible queries in the training set, we need to come up with a solution to predict the mixing proportions $P(z|Q_t;\mu)$ for any unseen queries that do not belong to the training collection. To generalize the parameters to new documents, Hofmann [Hof99] suggested a "fold-in" process for the latent class model by re-learning all training documents with the new document to generate an updated parameter estimation. However, this "fold-in" process by plugging in new queries and re-estimating the entire model is not reasonable in our task, because it requires a long time to process and more importantly, we do not have any relevance judgment for new queries to learn from.

To address this problem, we propose an adaptive approach aiming at parameterizing the mixing proportion $P(z|Q_t;\mu)$ using a specific set of features directly extracted from query topics, or called query features. They are able to capture important characteristics of users' information need. Formally, we can represent each query as a bag of query features $\{q_1, ...q_L\}$. The mixing proportions $P(z_k|Q;\mu)$ can then be modeled using a soft-max function $\frac{1}{Z}\exp(\sum_l \mu_{zl} q_l)$, where $Z = \sum_z \exp(\sum_l \mu_{zl} q_l)$ is the normalization factor that scales the exponential function to be a probability distribution. By substituting the mixing proportion back into Eqn(5.3), the BpLQA model can be rewritten as,

$$P(y_+|Q,D) = \frac{1}{Z}\sum_z \exp(\sum_l \mu_{zl} q_l)\sigma\left(\sum_{i=1}^{N}\lambda_{zi} f_i(Q,D)\right). \qquad (5.4)$$

Note that, since μ_{zl} is associated with each query feature instead of each training query, this modification allows the estimated μ_{zl} to be applied in any unseen queries as long as they can be formulated as vectors of query features. In the following discussions, we refer the model expressed in Eqn(5.4) to as the *adaptive pLQA*(**ApLQA**) model.

The log-likelihood function of ApLQA can be derived as follows,

$$l(\mu,\lambda) = \sum_{t=1}^{M_Q}\sum_{j=1}^{M_D}\log\left[\sum_z P(z|Q_t;\mu)\sigma\left(\sum_{i=1}^{N} y_{tj}\lambda_{ki} f_i(Q_t,D_j)\right)\right].$$

For this model, the EM algorithm can be derived similarly except the mixing proportions need to be substituted with Eqn(5.3.3). Therefore, the E-step computes the posterior probability of latent variable z given Q_t and D_j as follows,

$$h_{tj}(z) = \frac{\exp(\sum_l \mu_{zl} q_{tl})\sigma(\sum_i y_{tj}\lambda_{zi} f_i(Q_t,D_j))}{\sum_z \exp(\sum_l \mu_{zl} q_{tl})\sigma(\sum_i y_{tj}\lambda_{zi} f_i(Q_t,D_j))}.$$

In the M-step, we have the same update rule for updating λ_{zi}. For updating μ, we have

$$\mu_{zl} = \arg\max_{\mu_{zl}} \sum_{zt}\left(\sum_j h_{tj}(z)\right)\log\left[\frac{1}{Z_t}\exp(\sum_{l=1}^{L}\mu_{zl} q_{tl})\right],$$

where Z_t is the normalization factor for query Q_t, i.e., $Z_t = \sum_z \exp(\sum_l \mu_{zl} q_{tl})$. The M-step can be optimized by any gradient descent method. Note that, this step is actually fitting a multi-class logistic regression model with query features as inputs. In more detail, it is looking for the most similar logistic regression model with respect to the posterior probabilities of z for query Q_t.

The only remaining issue for defining the ApLQA model is to design a set of predictive query features. There are two useful principles to guide the design of suitable query features: 1) they should be able to be automatically generated from query descriptions (or from the statistics of ranking features), and 2) they should be predictive to estimate which latent classes the query belong to. For example, we can consider the presence/absence of specific person names in query topics, and the mean retrieval scores of each retrieval source as query features. There exists a trade-off for choosing the number of query features. Introducing more query features can represent the information need more precisely, but meanwhile the learning process will also demand more training data to achieve a stable estimation. It is still an open research direction to choose the optimal number of query features. Note that, in contrast to creating query classes that must be mutually exclusive, defining query features is much more flexible, eliminating the need to partition the query space into non-overlapping regions. Moreover, the number of query features can be much larger than the number of query classes with the same amount of training data.

Remark 1: The methodology of ApLQA follows a "divide-and-conquer" scheme with switches derived from query features. It handles the retrieval task by first dividing the input space (query feature space) into a set of regions and accordingly fitting specific decisions to the data in these regions. The regions have "soft" boundaries which allow data fall simultaneously in multiple regions, and the weights of combining experts are adjusted by learning algorithm. A gating function is used to produce scalar outputs as the linear weights for each expert and combines the probabilistic outputs of experts. ApLQA with query features has some similar properties with the mixture-of-expert architecture [JJ94], but it tailors itself to the retrieval tasks by separately using query features to control the choice of query mixture and using outputs from retrieval sources to generate the experts' decisions.

Remark 2: Why should we distinguish ranking features and query features? In short, this is because their design purposes differ with each other: query features are designed to control the selection of combination functions while ranking features are used to be the basic combination units for predicting document relevance. On the other hand, if we merge both ranking features and query features into a single logistic regression model, then the retrieval model needs to be rewritten as,

$$P(y_+|Q,D) = \sigma\left(\sum_i \lambda_i f_i(Q,D) + \sum_l \mu_l q_l(Q)\right). \tag{5.5}$$

Since λ and μ are not related to the query topics, the above model is essentially a query-independent combination model, which cannot take any advantages out of the query information.

Remark 3: Why do we need an intermediate latent class layer to capture the query information? Indeed, introducing this intermediate layer is not necessary. Recalling that our basic relevance-based retrieval model is of the following form,

$$P(y_+|D,Q) = \sigma\left(\sum_{i=0}^{N} \lambda_i(Q) f_i(D,Q)\right), \tag{5.6}$$

we can substitute the $\lambda_i(Q)$ to be a function associated with query features $q_l, l = 1...L$, that is,

$$P(y_+|D,Q) = \sigma\left(\sum_{i=0}^{N} \bar{\lambda}_i(q_1,...,q_L) f_i(D,Q)\right). \tag{5.7}$$

Now the retrieval task boils down to determine the function form of the combination weight $\bar{\lambda}_i$ and estimate the corresponding parameters. Unfortunately, so far it is yet not clear to us which function family could be a reasonable choice with a small number of parameters and a sufficient capability to represent the combination weights. A natural consideration is to represent $\bar{\lambda}_i$ as a linear function of $\sum_l \mu_{il} q_l(Q)$. However, limiting the combination weights to a linear function seems to be rather inflexible in practice. More importantly, this representation has $N \cdot L$ unknown parameters which is much higher than the number of parameters in ApLQA, $M_z N + M_z L$. According to these analysis, it turns out to be better off to introduce an intermediate query class layer into the model rather than directly modeling the combination parameters.

5.3.4 Kernel pLQA

By introducing explicit query features into the combination function, ApLQA can handle unseen queries that do not appear in the training data. However, the assumption of linear query feature combination in ApLQA might not be the best choice in general because the number of query features is often limited. Also, there exists some useful query information that cannot be described by explicit query feature representation. For example, the edit distance between two queries is a helpful hint for combination but it cannot be easily represented as a explicit query feature. Therefore, we develop an extension of the ApLQA model called the *kernel pLQA*(**KpLQA**) model that lends itself to the use of implicit feature space via Mercer kernels based on the representer theorem [KW71]. This extension is also motivated by a significant body of recent work that has demonstrated kernel methods are effective in a variety of applications.

In more detail, the kernel representation allows simple learning algorithms to construct a complex decision boundary by projecting the original input space to a high dimensional feature space, even infinitely dimensional in some cases. This seemingly computationally intensive task can be easily achieved through a positive definite reproducing kernel K and the well-known "kernel trick". To begin, let us rewrite the sum term $\sum_l \mu_{zl} q_l$ in Eqn(5.4) to be an arbitrary function $f_z(Q)$ with respect to the query Q,

$$P(y_+|Q,D) \propto \sum_z \exp f_z(Q) \cdot \sigma\left(\sum_{i=1}^{N} \lambda_{zi} f_i(Q,D)\right). \tag{5.8}$$

Since the loss function above only depends on the value of $f(\cdot)$ at the data points $\{f_z(Q)\}$, according to the representer theorem, the minimizer $f(x)$ admits a representation of the form $f_z(Q) = \sum_{k=1}^{M_D} \alpha_{zk} K(Q, Q_k)$, where $\{Q_k\}$ is the set of training queries, α_{zk} is the kernel fusion parameter for z and Q_k, and $K(\cdot,\cdot)$ is a Mercer kernel on the query space which has to be positive definite. With this kernel representation, we can derive the corresponding log-likelihood function by substituting original mixing proportion term $P(z|Q_t)$ to be,

$$P(z|Q_t) = \frac{1}{Z} \exp(\sum_{k=1}^{M_D} \alpha_{zk} K(Q_t, Q_k))$$

ApLQA is a special case of KpLQA if each element of K is chosen to be the inner product between the query features of two queries. However, the flexibility of kernel selection has offered more powers to the KpLQA model. For example, the kernel function can have different forms such as the polynomial kernel $K(u,v) = (u \cdot v + 1)^p$ and the Radial Basis Function (RBF) kernel $K(u,v) = \exp(-\gamma \|u - v\|^2)$. The latter one has the ability to project the query features into an infinite dimensional feature spaces. Moreover, we can transform the distance metric between queries (e.g., edit distance between queries) into the implicit feature space in form of a Mercer kernel, instead of designing explicit features for each query. Also, it is possible to linearly combine multiple Mercer kernels obtained from various places to form another positive definite Mercer kernel.

The optimization of KpLQA is similar as that of ApLQA except some slight changes in the mixing proportion terms of the E-step and M-step. Formally, we can rewrite the E-step to be the following,

$$h_{kj}(z) = \frac{\exp(\sum_k \alpha_{zk} K_{kt}) \sigma(\sum_i y_{tj} \lambda_{zi} f_i(Q_t, D_j))}{\sum_z \exp(\sum_k \alpha_{zk} K_{kt}) \sigma(\sum_i y_{tj} \lambda_{zi} f_i(Q_t, D_j))}.$$

In the M-step, we have the same update rule for updating λ_{zi} but for updating α_{zk}, we have

$$\alpha_{zk} = \arg\max_{\alpha_{zk}} \sum_{zk} \left(\sum_j h_{kj}(z) \right) \log \left[\frac{1}{Z_t} \exp(\sum_{k=1}^{M_D} \alpha_{zk} K_{kt}) \right],$$

where $K_{kt} = K(Q_t, Q_k)$, Z_t is the normalization factor for query Q_t, i.e., $Z_t = \sum_z \exp(\sum_k \alpha_{zk} K_{kt})$. Similarly, the M-step can be optimized by any gradient descent methods.

5.3.5 Hierarchical pLQA Model

The pLQA model assumes that the combination parameters of each training query can be completely represented by the parameters of a latent query class λ_z. However, this model assumption is not always sensible because specific queries usually contain outlier noises beyond a limited number of latent query classes. Therefore, it might be worthwhile to explicitly model the query-specific variation in the phase of parameter estimation. To be more precise, we can express the combination parameters of the current query ω as the sum of two terms,

$$\omega = \lambda_z + \epsilon,$$

where λ_z is the deterministic combination parameters corresponding to the z^{th} query class and ϵ is a random variable drawn from Gaussian distribution $\mathcal{N}(0, \Sigma)$ that captures the query-specific variation in a single query class. Therefore, ω is a random variable drawn from a Gaussian distribution with mean λ_z and variance σ^2. Based on this construction, the sampling process of document relevance Y can be viewed as the following,

1. Choose a query class $z \sim$ Multinomial(μ) for query Q,

2. Choose a weight $\omega_i \sim N(\lambda_{zi}, \Sigma)$,

3. Choose a response $Y \sim$ Bernoulli(p), where $logit(p) = \sum_i \omega_i f_i(D, Q)$,

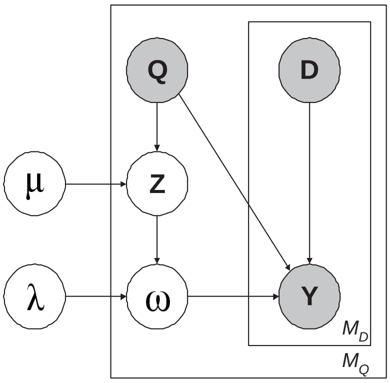


Figure 5.5: Graphical model representation for the hierarchical pLQA model.

This can be formulated as a hierarchical Bayesian model, of which the graphical model representation is shown in Figure 5.5. Accordingly, the joint probability of relevance y, latent variable z and combination parameters ω can be written as,

$$P(y_+, \omega, z|Q, D; \mu, \lambda) \;\; = \;\; P(z|Q, \mu)P(\omega|z, \lambda)P(y|Q, D, \omega) \qquad (5.9)$$

By marginalizing out the ω in the joint probability, the corresponding conditional probability of relevance can be written as,

$$\begin{aligned} P(y_+|Q, D; \mu, \lambda) &= \sum_z \int_\omega P(z|Q, \mu)P(\omega|z, \lambda)P(y|Q, D, \omega) \\ &= \sum_z \int_\omega d\omega P(z|Q, \mu)P(\omega|z, \lambda) \cdot \sigma\left(\sum_{i=1}^N \omega_i f_i(D, Q)\right). \end{aligned}$$

In this thesis, we call above retrieval model the *hierarchical pLQA model*. Note that when the variance of distribution $P(\omega|z, \lambda)$ goes to 0, the hierarchical pLQA model will naturally degrade to the pLQA model described in Eqn(5.3). The optimal parameters can be estimated by maximizing the following log-likelihood function,

$$L(\mu, \lambda) = \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} \log \left[\sum_z \int_\omega d\omega P(z|Q_t; \mu)P(\omega|z; \lambda) \cdot \sigma\left(\sum_{i=1}^N \omega_i f_i(Q_t, D_j)\right)\right]$$

Remark: We can view the hierarchical pLQA model from another perspective. By substituting the distribution of ω with a point estimation ω_t^* for each query q_t, we can rewrite the log-likelihood as follows,

$$
\begin{aligned}
L(\mu, \lambda, \omega^*) &= \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} \log \left[\sum_z P(z|Q_t) P(\omega_t^*|z) \cdot \sigma \left(\sum_{i=1}^{N} \omega_{ti}^* f_i(Q_t, D_j) \right) \right] \\
= \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} \log & \left[\sum_z P(z|Q_t) P(\omega_t^*|z) \right] + \sum_{t=1}^{M_Q} \sum_{j=1}^{M_D} \log \left[\sigma \left(\sum_{i=1}^{N} \omega_{ti}^* f_i(Q_t, D_j) \right) \right],
\end{aligned}
$$

where the distribution of ω_i^* given each class z_k is a normal distribution $N(\lambda_{ki}, \Sigma)$. If we approximate the optimization of $L(\mu, \lambda, \omega^*)$ in Eqn(5.10) by sequentially maximizing the latter logistic regression term followed by maximizing the former term, we can derive a two-staged learning process as follows. First, we start with the maximum likelihood estimation for the combination parameters ω_{ti}^* for each query q_t using logistic regression. It follows by estimating a latent semantic model [Hof99] on the space of ω^*, of which the goal is to discover the latent mixing structure of the pair between weights ω^* and queries Q_t. This provides an intuitive interpretation for the hierarchical pLQA model, where the optimization of hierarchical pLQA model can be thought of as a process to extract the latent causes from the best-fit query-specific weights ω^* of all training queries. Moreover, this formulation offers a more convenient method to predict the combination parameters ω of unseen queries which is able to circumvent the unaffordable integral operation on ω.

Inference

Unfortunately, the exact inference in the hierarchical pLQA model is intractable. Therefore we usually resort to some approximate inference approaches such as *variational methods* to optimize the objective function. Specifically, we adopt the mean field approximation [PA87] in our derivation, where the basic idea is to use a tractable family of distributions $q(z, \omega)$ to approximate the true conditional distribution $p(z, \omega|y, Q, D)$ which is unable to estimate directly. Usually, the approximate distribution $q(z, \omega)$ decouples all the hidden variables of z, ω and makes them become independent with each other. In more detail, we construct the family of variational distributions as follows,

$$
q(z, w) = q(w|\gamma, V) q(z|\phi),
$$

as a surrogate to approximate the posterior distribution $p(z, w|y_j, Q, D_j)$, where $q(w|\gamma, V)$ is a Gaussian distribution with mean γ and the variance V, $q(z|\phi)$ is a multinomial distribution with K parameters ϕ_k. The independence between variables in the variational distributions results in an efficient inference algorithm as shown below. According to the Jensen's inequality, the optimization problem can be translated into maximizing a lower bound of the log-likelihood function. The variational lower bound is usually much simpler to optimize and it is equal to the original log-likelihood only when $q(z, w) = p(z, w|y_j, Q, D_j)$.

By taking derivatives with respect to each variational parameter, we can obtain the following coordination ascent algorithm:

1. Update the logistic variational parameter ξ_j for each document,

$$\xi_j = [x_j^T (V + \gamma\gamma^T) x_j]^{1/2}.$$

2. For each latent query class, optimize the multinomial parameter ϕ_k,

$$\phi_k \propto \mu_k \exp[-\frac{1}{2}(\lambda_k - \gamma)^T \Sigma^{-1} (\lambda_k - \gamma)].$$

3. Update the variational parameter for combination γ,

$$\gamma = \left[M_D \Sigma^{-1} - 2 \sum_j g(\xi_j) x_j x_j^T \right]^{-1} (\sum_k M_D \phi_k \Sigma^{-1} \lambda_k + \frac{1}{2} \sum_j y_j x_j).$$

$$V = \left(\Sigma^{-1} - \frac{2}{M_D} \sum_{i=1}^{n} g(\xi_j) x_i x_i^T \right)^{-1}.$$

These fixed point equations are invoked iteratively until the change of KL-divergence is small enough. Upon convergence, we use the resulting q as a surrogate to the original posterior p, and compute the approximate conditional probability of $p(y|Q, D)$. Please refer to the Appendix for more details of the derivation.

Parameter Estimation

For the purpose of parameter estimation, we need to handle a collection of training queries $Q_1, ..., Q_{M_Q}$. Formally we want to find the model parameter μ and λ that can maximize the log-likelihood of the collection,

$$L(\mu, \lambda) = \sum_{tj} \log p(y_{tj}|Q_t, D_j; \mu_t, \lambda).$$

By applying the variational approximation again, we can achieve a similar lower bound on the log-likelihood that can be maximized w.r.t. the model parameters. The optimal parameters can be obtained by iterating the variational Expectation-Maximization(EM) process. In the E-step, the lower bound L' are maximized w.r.t. the variational parameters until the convergence as described above. In the M-step, we maximize the lower bound w.r.t. the model parameters. This process tries to find the maximum likelihood estimates under the approximate posterior provided in the E-step. The overall process corresponds to a stepwise optimization in the lower bound L'.

To maximize the lower bound w.r.t. the latent query class parameters μ_{tk}, we can have

$$\{\mu_{kl}\} \quad = \quad \arg\max_{\mu_{kl}} \sum_{t=1}^{M_Q} \sum_{k=1}^{K} \phi_{tk} \log \left[\frac{1}{Z_t} \exp(\sum_{l=1}^{L} \mu_{kl} q_{tl}) \right],$$

which can solved by gradient descent methods. To maximize the bound w.r.t. the combination weight parameters λ, we can obtain its derivative as,

$$\frac{\partial L'}{\partial \lambda_k} = \sum_t \phi_{tk} \Sigma^{-1} (\lambda_k - \gamma_t).$$

Setting the derivative to be zero yields,

$$\lambda_k = \left(\sum_t \phi_{tk} \gamma_t \right) / \sum_t \phi_{tk}.$$

5.3.6 Geometric Interpretation

In this section, we provide a geometric interpretation to highlight the differences between the models discussed in this chapter, i.e., query-independent combination, query-class combination, BpLQA and FpLQA/KpLQA. Fig. 5.6 illustrates their comparison on a combination parameter space with two features and the query-class simplex with three classes spanned by the mixture models.

All of these combination models are constructed on the space of combination parameters, where each set of combination weights corresponds to a point in the space. In a geometrical view, query-independent combination maps all the queries into a single point on the space and assume that every query should share the same

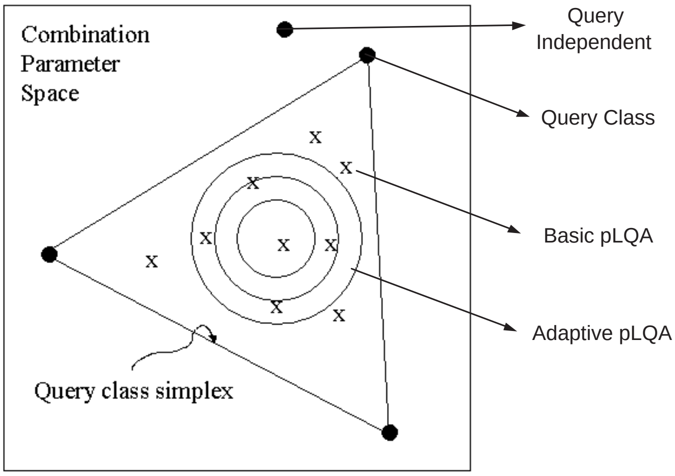


Figure 5.6: Illustration of the combination parameter space with two ranking features and the query-class simplex with three classes spanned by mixture models. The corners of the query-class simplex indicates that the conditional probability of one of the query classes are one and others are zero. Each point in the simplex also corresponds to a set of different combination parameters. The query independent combination considers only one point in the parameter space. The query-class combination selects parameters from one of the corners on the simplex. BpLQA learns an empirical latent query-class distribution inside the simplex marked by "x". ApLQA provides a smooth distribution on the query features marked by the contour line.

combination weights. However, the other three mixture models project each query into a point on a $M_z - 1$ dimensional simplex containing all possible multinomial distributions, called query-class simplex. Each point inside the simplex also corresponds to a set of different combination parameters. But these mixture models make use of the query-class simplex in a different ways. The query-class combination model only considers selecting combination parameters from one of the corners on the simplex for each query. BpLQA operated in a flexible manner which can learn the query-class distributions inside the simplex, however, since the query-class distributions are specific to the training query itself, the learned parameters cannot be extended to other queries. ApLQA provided a smoothed query-class distribution based on the query features and thus it is able to apply on both observed and unseen queries.

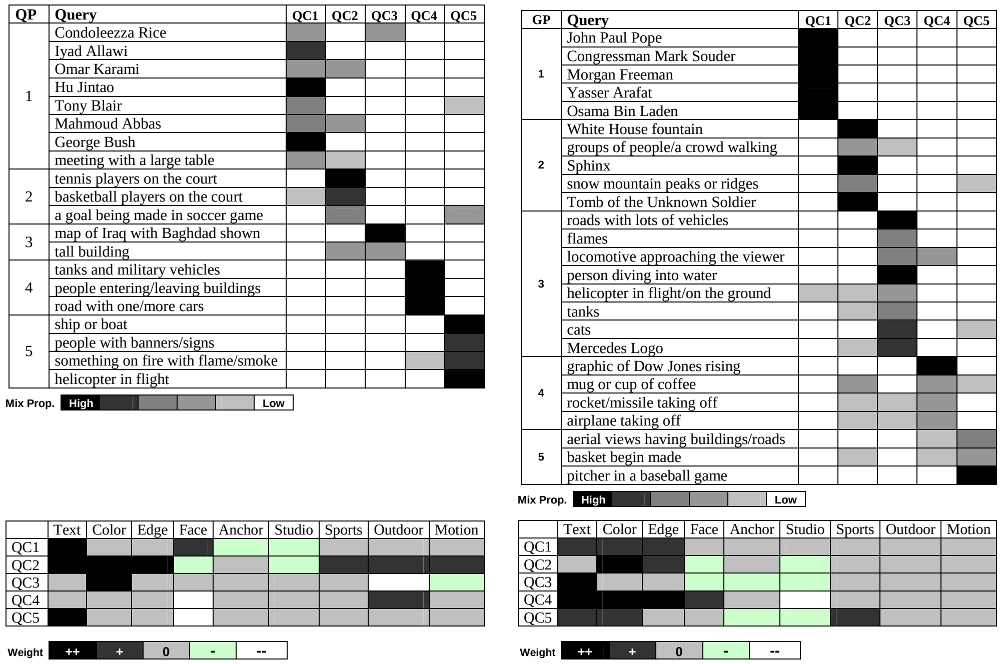
5.4 Experiments

The experimental settings of our following experiments is similar to the ones described in Section 5.2. To initialize the EM algorithm in pLQA, we randomly set the mixing parameters μ_z. The combination parameters λ_z are initialized from M_z individual queries that are automatically selected using a simple heuristic similar to the maximal margin relevance. In order to implement the ApLQA and KpLQA model, we also designed the following binary query features,

- Named persons: does the query contain specific person names;
- Named objects: does the query contain specific object names;
- Multiple objects: is there more than two objects mentioned in the query;
- People: does the query require the appearance of people/crowd;
- Sports: is the query related to sport events;
- Vehicle: is the query related to vehicle;
- Movement: does the query involve moving objects;
- Query difficulty: does the text retrieval return more than 100 documents;
- Similarity of image examples: are the image examples similar with each other in terms of color or texture;
- Image face: does the image contain faces.

All these query features are useful to predict the combination function for unseen queries and they can be automatically detected from the query description through some manually defined rules plus natural language processing and image processing techniques. For example, the first three query features can be obtained by applying named entity extraction, NP tagging and shallow parsing, of which the details can be found in Section 5.1[3]. The features of sports, vehicle and people can be detected using a manually defined vocabulary with a limited set of words. The eighth and ninth query features can be automatically generated by existing image processing and face detection techniques. Finally the last query feature can

[3] This section also provides evidence that above query features can be predicted with an accuracy above 93%, which is sufficient to support the following retrieval process.



QP	Query	QC1	QC2	QC3	QC4	QC5
1	Condoleezza Rice	░		░		
	Iyad Allawi	▓				
	Omar Karami	░	░			
	Hu Jintao	█				
	Tony Blair	▒				□
	Mahmoud Abbas	▒	░			
	George Bush	█				
	meeting with a large table	░	□			
2	tennis players on the court		█			
	basketball players on the court	□	▓			
	a goal being made in soccer game		▒			░
3	map of Iraq with Baghdad shown			█		
	tall building		░	░		
4	tanks and military vehicles				█	
	people entering/leaving buildings				█	
	road with one/more cars				█	
5	ship or boat					█
	people with banners/signs					▓
	something on fire with flame/smoke				□	▓
	helicopter in flight					█

Mix Prop.	High					Low
	█	▓	▒	░	□	

	Text	Color	Edge	Face	Anchor	Studio	Sports	Outdoor	Motion
QC1	++	0	0	+	-	-	0	0	0
QC2	++	++	++	-	0	-	+	+	+
QC3	0	++	0	0	0	0	0	--	-
QC4	0	0	0	--	0	0	0	+	0
QC5	++	0	0	--	0	0	0	0	0

Weight	++	+	0	-	--

(a)

GP	Query	QC1	QC2	QC3	QC4	QC5
1	John Paul Pope	█				
	Congressman Mark Souder	█				
	Morgan Freeman	█				
	Yasser Arafat	█				
	Osama Bin Laden	█				
2	White House fountain		█			
	groups of people/a crowd walking		░	□		
	Sphinx		█			
	snow mountain peaks or ridges		▒			□
	Tomb of the Unknown Soldier		█			
3	roads with lots of vehicles			█		
	flames			▒		
	locomotive approaching the viewer			▒	░	
	person diving into water			█		
	helicopter in flight/on the ground	□	□	░		
	tanks		□	▒		
	cats			▓		□
	Mercedes Logo		□	▓		
4	graphic of Dow Jones rising				█	
	mug or cup of coffee		░		░	□
	rocket/missile taking off		□	□	░	
	airplane taking off		□	□	░	
5	aerial views having buildings/roads				□	▒
	basket begin made		□		□	░
	pitcher in a baseball game					█

Mix Prop.	High					Low
	█	▓	▒	░	□	

	Text	Color	Edge	Face	Anchor	Studio	Sports	Outdoor	Motion
QC1	+	+	+	0	0	0	0	0	0
QC2	0	++	+	-	0	-	0	0	0
QC3	++	0	0	-	-	-	0	0	0
QC4	++	++	++	+	0	--	0	0	0
QC5	+	+	0	0	-	-	+	0	0

Weight	++	+	0	-	--

(b)

Figure 5.7: Higher: we organize (a) TREC'05 and (b) TREC'03 queries into five groups (QP) based on which query class (QC) has the highest mixing proportions. The left two columns show group IDs and queries. The other columns indicate the query's mixing proportions w.r.t. five latent classes discovered by BpLQA where darker blocks mean higher value. Lower: the combination weights of selected ranking features for each latent class. Similarly, the darker the higher value.

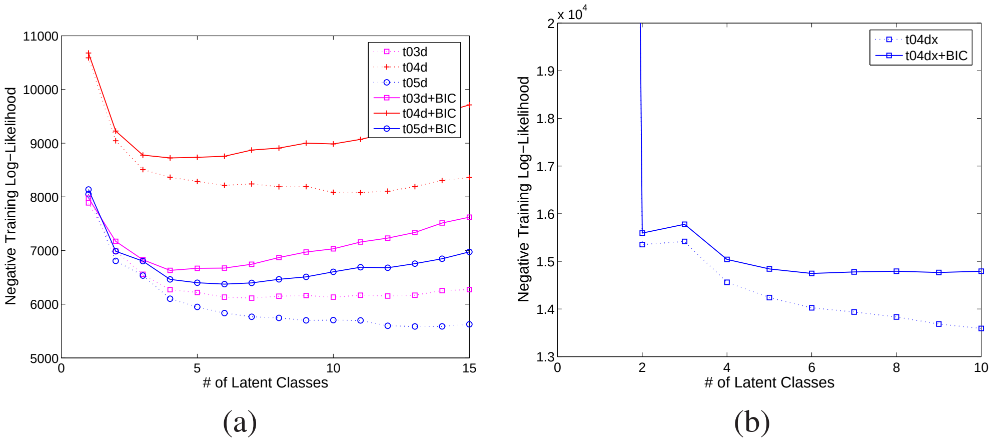
be simply obtained from text retrieval statistics, where we judge a query to be difficult if the system only returns less than 100 documents. Note that our current implementation requires some query features to be generated from manually defined rules. It is an interesting direction to explore fully automatic approaches to extract query features but we leave it as our future work.

5.4.1 Illustration of latent query classes

To illustrate the ability of pLQA to discover meaningful latent query classes, Figure 5.7(a) and Figure 5.7(b) show five latent query classes and their corresponding combination weights that are automatically learned from the BpLQA model on the TREC'05 and TREC'03 development data. In this figure, all of the queries are listed except those without any training data in the development set. The blocks on the right show the scale of mixing proportions $p(z|Q)$ of each query Q w.r.t. each query class z. These queries are organized into five groups based on the class IDs of their maximal mixing proportion in all query classes, i.e., $\arg\max_z p(z|Q)$. In the lower figure, we plot the combination weights of selected ranking features for each class.

It is interesting to examine whether the query grouping suggested by BpLQA are sensible. Among the five groups of the TREC'05 queries, the first one mainly contains all the "named person" queries in the training data and one query related to person appearance, "meeting with a large table". This group of queries usually has a high retrieval performance when using the text features and prefers the existence of person faces, while content-based image retrieval is not effective for them. The second group consists of three queries related to sport events including "basketball", "soccer" and "tennis". These queries often rely on both text retrieval and image retrieval results, because these sport scenes in the news broadcast usually share common transcript words and image backgrounds. Moreover, their search results can be improved by utilizing the detection of semantic concepts of sports, outdoors and motion. The last three groups all contain queries related to object finding, however, they have many distinctions in the combination strategies. In the third group, the queries tend to search for objects that have similar visual appearances without any apparent motions. The case of the fourth group is quite different. These queries are mainly looking for objects in the outdoors scene such as "road" and "military vehicle". Finally, the fifth group seems to be a general group that contains all remaining queries. The queries in this group search for objects without visual similarities and thus place a high weight on the text retrieval since text retrieval is usually the most reliable retrieval component in general. Similarly, the automatic grouping of the TREC'03 queries also sheds light on the clustering pattern of its query space. For example, the first group corresponds to the queries of finding specific persons which can benefit from both text retrieval and image retrieval. The second group is mainly related to the queries of finding specific objects where image retrieval can contribute the most.

Most of the discovered latent query classes are consistent with the observations made by many previous studies [YYH04, CNL$^+$04], such as the classes of named person, objects and sports. Meanwhile, BpLQA is also able to suggest some query classes that have not been suggested before such as the class of "outdoor objects". This analysis shows that BpLQA can achieve a reasonable query classification through a completely data-driven statistical learning approach instead of a manual



(a) (b)

Figure 5.8: Negative training log-likelihood of BpLQA against the number of query classes for the collections of (a) t03/04/05d and (b) t04dx. The dash lines indicate the original log-likelihood and the solid lines indicate the regularized log-likelihood with BIC.

handcrafting procedure.

Finally, it is also helpful to analyze the query mixing patterns learned from BpLQA. For example, the query "Omar Karami" can be described by a mixture of the first and the second query classes. It is not obvious why the second query class is related at first sight. But after a further analysis, we found that Karami always showed up with a similar background when he was meeting foreigners and hence visual appearance turned out to be a helpful clue to find him. Such a mixture treatment provides more flexibility in retrieval modeling and offers deeper understanding of the queries.

5.4.2 Retrieval Results

Next we present the multimedia retrieval results on all the search sets using the proposed models and parameters learned from the development set. Since BpLQA cannot generalize its parameters to unseen queries, we use the development set on the same year as the search set to estimate its parameters. For example, the parameters estimated on the set $t03d$ is reused in the search set $t03s$. For ApLQA and KpLQA, there is no such a constraint and therefore the parameters are estimated with a larger training set $t04dx$.

As discussed before, we can obtain the number of query classes by optimiz-

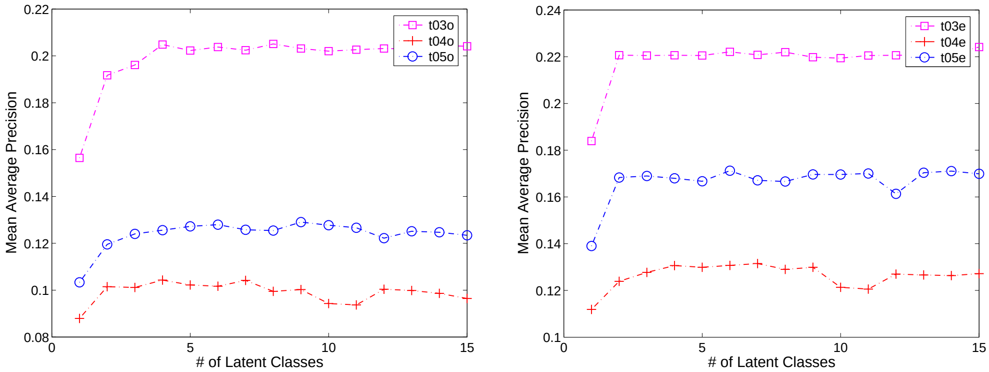


Figure 5.9: Performance of BpLQA vs. the number of latent query classes.

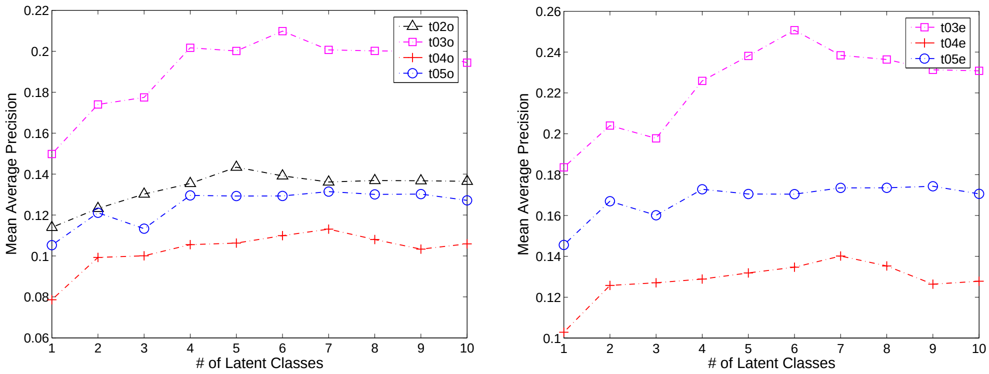


Figure 5.10: Performance of ApLQA vs. the number of latent query classes.

ing the regularized log-likelihood with an additional BIC term. Figure 5.8(a) and Figure 5.8(b) plot the learning curves of the negative log-likelihood and their regularized counterparts on four development sets (i.e., t03d, t04d, t05d and t04dx) against the number of query classes. It can be observed that when the number of query classes grows, the learning curves become consistently lower and lower until they asymptotically reach a saturated level. Occasionally the curves even slightly raise at the end, which shows the training data seem to be overfitted as the number of query classes become larger. Based on the statistical model selection principle, the number of query classes for each training set can be determined by seeking the lowest point on the regularized log-likelihood with an additional BIC term. Given the learning curves, we can find that the optimal numbers of latent classes turn out to be 4, 4, 6 and 6 for the collections of t03d, t04d, t05d and t04dx

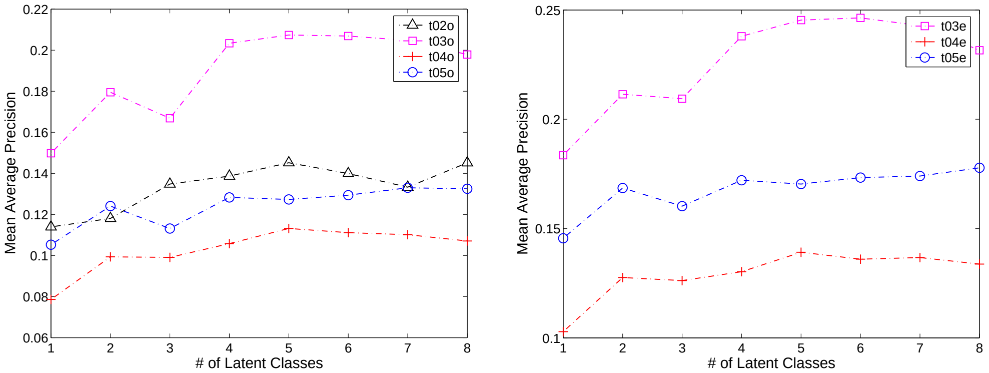


Figure 5.11: Performance of KpLQA with the RBF kernel vs. the number of latent query classes.

respectively.

To examine whether the prediction of optimal class number is reasonable and whether the number of query classes affects the retrieval performance on the search sets, we depict the BpLQA performance curve in Figure 5.9 on six search sets with the number of query classes growing from 1 to 15. As can be seen, the MAPs increase dramatically with more query classes when the class number is under four. For example, in $t03o$ the MAP is boosted from 15% when using one class to 21% when using four classes. But, afterwards the performance can only be improved slightly with a higher number of classes and even be degraded at times, which indicates the retrieval performance cannot always benefit from incorporating additional query classes. Another important observation is that the MAP achieved by using the predicted number of query classes are always among the highest points in the entire curve. We also plot the performance curves of ApLQA in Figure 5.10, KpLQA with the RBF kernel in Figure 5.11 and HpLQA with $\sigma^2 = 100$ in Figure 5.12 with the number of query classes growing from 1 to 15. The model parameters are learned from the development set of $t04dx$. Similar observations can be obtained from these results, where the retrieval performance usually grows in a significant amount at the beginning of the curves and then drop a little bit at the end (although HpLQA does not seem to be overfitting yet at the setting of 15 query classes). In all these graphs, the estimated number of latent query classes, i.e., 6 query classes, is among one of the best configurations for pLQA. This verified the effectiveness of learning the optimal number of query classes based on the proposed pLQA models.

In order to study the sensitivity of pLQA with respect to the number of train-

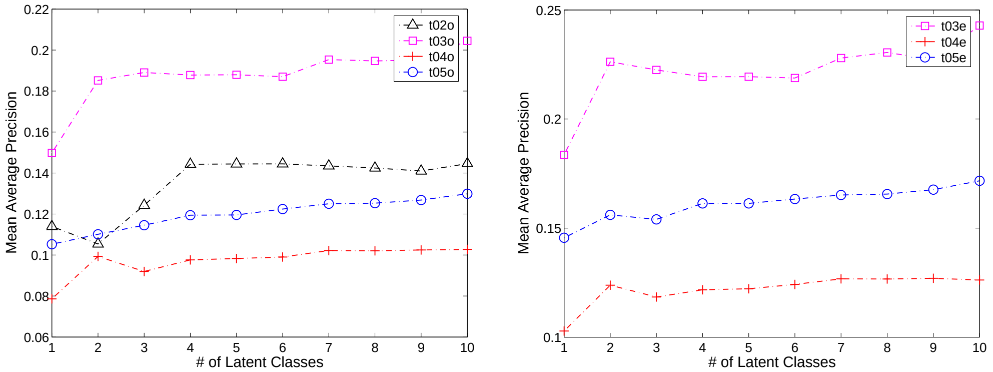


Figure 5.12: Performance of HpLQA with $\sigma^2 = 100$ vs. the number of latent query classes.

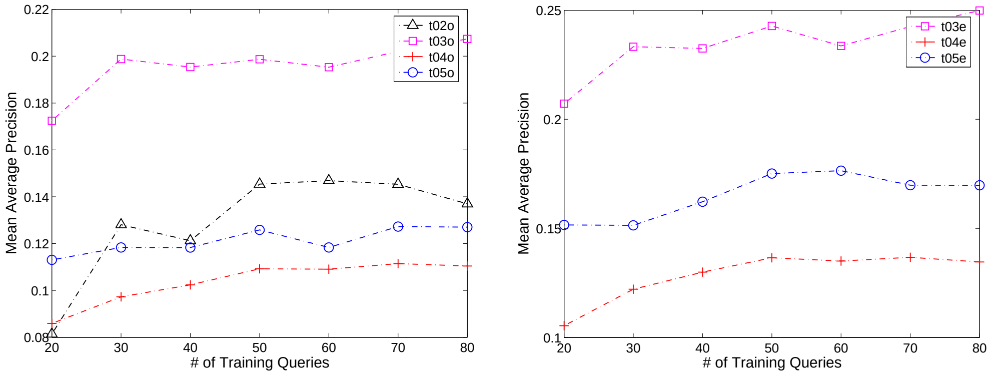


Figure 5.13: Performance of ApLQA vs. the number of training queries.

ing data, we varied the number of queries from 20 to 80 in the development set $t04dx$ and re-estimate the parameters for pLQA models. Figure 5.13 shows the learning curve of ApLQA in terms of mean average precision against the number of training queries. As expected, the proposed ApLQA model usually benefits from incorporating more and more training data until it approaches an asymptote. In this setting (with 6 latent query classes), the minimal number of queries that can achieve a near-asymptote performance is 70 for the $t03o/e$ collections and 50 for the other collections.

Table 5.6 lists a detailed comparison between BpLQA with the estimated number of query classes and several baseline methods including text retrieval (Text), query independent (QInd) and query-class combination (QClass) methods with

Data	Method	MAP	P30	P100	R1k	Person	SObj	GObj	Sport	Other
t03o	Text	0.146(+0%)	0.171	0.118	0.477	0.371	0.230	0.068	0.031	0.007
	QInd	0.150(+2%)	0.212	0.136	0.520	0.251	0.293	0.095	0.101	0.012
	QClass*	0.173(+19%)	0.207	0.129	0.531	0.371	0.307	0.091	0.088	0.009
	BpLQA**	**0.205(+40%)**	0.241	0.145	0.565	0.450	0.361	0.102	0.100	0.011
t03e	Text	0.192(+0%)	0.231	0.155	0.564	0.402	0.296	0.103	0.134	0.044
	QInd	0.184(-4%)	0.243	0.180	0.591	0.260	0.311	0.128	0.289	0.027
	QClass*	0.212(+10%)	0.264	0.172	0.594	0.402	0.306	0.125	0.265	0.044
	BpLQA**	**0.221(+14%)**	0.283	0.184	0.628	0.387	0.358	0.132	0.256	0.045
t04o	Text	0.078(+0%)	0.178	0.107	0.361	0.188	0.012	0.033	0.046	0.044
	QInd	0.079(+0%)	0.177	0.116	0.375	0.144	0.063	0.034	0.108	0.051
	QClass	0.087(+11%)	0.186	0.115	0.380	0.194	0.080	0.030	0.090	0.044
	BpLQA	**0.104(+33%)**	0.201	0.124	0.379	0.248	0.024	0.035	0.118	0.047
t04e	Text	0.097(+0%)	0.184	0.125	0.461	0.199	0.003	0.041	0.152	0.048
	QInd	0.103(+5%)	0.191	0.144	0.468	0.157	0.006	0.036	0.232	0.072
	QClass	0.111(+14%)	0.217	0.143	0.469	0.206	0.005	0.035	0.221	0.064
	BpLQA	**0.131(+34%)**	0.229	0.156	0.473	0.262	0.009	0.035	0.243	0.069
t05o	Text	0.073(+0%)	0.207	0.175	0.339	0.141	0.015	0.097	0.075	0.016
	QInd	0.105(+44%)	0.268	0.205	0.387	0.164	0.029	0.090	0.271	0.017
	QClass*	0.108(+47%)	0.261	0.200	0.396	0.171	0.032	0.061	0.309	0.018
	BpLQA**	**0.128(+75%)**	0.307	0.212	0.388	0.212	0.039	0.100	0.316	0.018
t05e	Text	0.103(+0%)	0.253	0.210	0.389	0.178	0.038	0.095	0.181	0.029
	QInd	0.146(+40%)	0.354	0.251	0.428	0.218	0.070	0.086	0.395	0.032
	QClass*	0.147(+42%)	0.329	0.248	0.437	0.219	0.081	0.055	0.435	0.033
	BpLQA**	**0.171(+65%)**	0.382	0.269	0.439	0.273	0.089	0.097	0.445	0.029

Table 5.6: Comparison of combination methods. The numbers in brackets show the improvement over Text. Bold texts indicate the highest performance in each criterion. * means statistical significance over Text with *p-value* < 0.01 (sign tests) and ** means significance over both Text and QInd.

five classes defined in [YYH04]. QClass has shown to be one of the best overall retrieval systems in the past TRECVID evaluations. The parameters in all baseline methods were learned using the same training sets as BpLQA. All the results results are reported in terms of the mean average precision(MAP) up to 1000 documents, precision at top 30, 100 documents and recall at top 1000 documents. To analyze the results in more detail, we also grouped the queries in each collection and reported their MAPs in five different categories, i.e., named person, special object, general object, sports and general queries. As can be observed from the results, QClass is usually superior to both QInd and Text. On average, it brings a roughly 3% absolute improvement (or 30% relative improvement) over the text retrieval. As compared to QClass, BpLQA achieves another 2% boost in terms of MAP without any manual tuning on the query class definitions. As it turns out, the differences between BpLQA and Text/QInd become statistically significant in two out of three collections. The major performance growth factor for BpLQA

Data	Method	MAP	P30	P100	R1k	Person	SObj	GObj	Sport	Other
t02o	Text	0.098(+0%)	0.113	0.076	0.421	0.124	0.183	0.079	0.023	0.011
	QClass*	0.132(+34%)	0.131	0.086	0.425	0.316	0.207	0.075	0.004	0.013
	ApLQA*	0.139(+41%)	0.136	0.084	0.454	0.364	0.194	0.078	0.014	0.017
	KpLQA-R*	0.140(+42%)	0.139	0.084	0.450	0.365	0.201	0.075	0.014	0.017
	KpLQA-P*	**0.144(+46%)**	0.139	0.083	0.462	0.388	0.206	0.070	0.023	0.019
t03o	Text	0.146(+0%)	0.171	0.118	0.477	0.371	0.230	0.068	0.031	0.007
	QClass*	0.200(+37%)	0.236	0.137	0.542	0.407	0.336	0.088	0.106	0.015
	ApLQA*	**0.210(+44%)**	0.249	0.144	0.562	0.463	0.358	0.106	0.103	0.017
	KpLQA-R*	0.207(+42%)	0.248	0.143	0.565	0.469	0.340	0.107	0.100	0.015
	KpLQA-P*	0.206(+41%)	0.249	0.144	0.544	0.467	0.347	0.097	0.107	0.018
t03e	Text	0.192(+0%)	0.231	0.155	0.564	0.402	0.296	0.103	0.134	0.044
	QClass*	0.239(+24%)	0.289	0.187	0.612	0.409	0.345	0.127	0.293	0.036
	ApLQA*	0.246(+27%)	0.301	0.179	0.628	0.491	0.383	0.133	0.296	0.049
	KpLQA-R*	**0.246(+28%)**	0.300	0.188	0.627	0.493	0.370	0.137	0.291	0.033
	KpLQA-P*	0.240(+25%)	0.303	0.186	0.619	0.493	0.352	0.127	0.305	0.032
t04o	Text	0.078(+0%)	0.178	0.107	0.361	0.188	0.012	0.033	0.046	0.044
	QClass*	0.094(+20%)	0.199	0.125	0.381	0.194	0.080	0.046	0.108	0.045
	ApLQA*	0.110(+41%)	0.220	0.119	0.381	0.255	0.053	0.044	0.110	0.051
	KpLQA-R*	**0.111(+42%)**	0.212	0.120	0.379	0.262	0.067	0.040	0.108	0.050
	KpLQA-P	0.109(+40%)	0.213	0.128	0.380	0.255	0.030	0.037	0.116	0.055
t04e	Text	0.097(+0%)	0.184	0.125	0.461	0.199	0.003	0.041	0.152	0.048
	QClass	0.113(+16%)	0.212	0.152	0.470	0.206	0.005	0.046	0.233	0.056
	ApLQA*	0.135(+38%)	0.239	0.149	0.471	0.268	0.030	0.043	0.237	0.070
	KpLQA-R*	**0.136(+40%)**	0.241	0.150	0.469	0.275	0.032	0.041	0.238	0.069
	KpLQA-P*	0.135(+38%)	0.242	0.151	0.471	0.267	0.028	0.035	0.244	0.076
t05o	Text	0.073(+0%)	0.207	0.175	0.339	0.141	0.015	0.097	0.075	0.016
	QClass*	0.116(+58%)	0.292	0.211	0.402	0.173	0.031	0.100	0.322	0.017
	ApLQA*	0.129(+77%)	0.294	0.212	0.397	0.209	0.042	0.104	0.328	0.017
	KpLQA-R*	0.129(+77%)	0.307	0.221	0.396	0.207	0.040	0.108	0.327	0.018
	KpLQA-P*	**0.130(+78%)**	0.290	0.214	0.390	0.211	0.042	0.100	0.331	0.018
t05e	Text	0.103(+0%)	0.253	0.210	0.389	0.178	0.038	0.095	0.181	0.029
	QClass*	0.168(+62%)	0.376	0.265	0.445	0.258	0.079	0.096	0.452	0.033
	ApLQA*	0.170(+64%)	0.390	0.266	0.441	0.269	0.084	0.098	0.452	0.030
	KpLQA-R*	0.173(+67%)	0.386	0.270	0.445	0.268	0.097	0.104	0.450	0.032
	KpLQA-P*	**0.173(+67%)**	0.394	0.276	0.436	0.272	0.083	0.096	0.465	0.033

Table 5.7: Comparison of text retrieval, QClass, ApLQA and KpLQA.

can be traced to a higher precision on the top documents, although the recall numbers of all these methods do not vary a lot. By comparing MAPs with respect to each query type, we find that BpLQA benefits most from the Person and Special Object type queries, as well as the General Object type in $t05s$. This is because these query types have their information needs clearly defined and thus they are able to be improved by making better use of the training data.

Table 5.7 compares text retrieval and query-class combination (QClass-x) with ApLQA, KpLQA using the RBF kernel with $\gamma = 0.01$ (KpLQA-R), KpLQA using the polynomial kernel with $p = 3$ (KpLQA-P). All the parameters are estimated from the external training set $t04dx$. Each pLQA model is learned with six

Data	Method	MAP	P30	P100	R1k	Person	SObj	GObj	Sport	Other
t02o	Text	0.098(+0%)	0.113	0.076	0.421	0.124	0.183	0.079	0.023	0.011
	ApLQA	0.139(+41%)	0.136	0.084	0.454	0.364	0.194	0.078	0.014	0.017
	HpLQA-V100	0.142(+44%)	0.149	0.094	0.468	0.389	0.196	0.071	0.006	0.024
	HpLQA-V50	0.145(+47%)	0.140	0.090	0.470	0.389	0.206	0.076	0.003	0.016
	HpLQA-V30	**0.146(+48%)**	0.156	0.090	0.474	0.389	0.214	0.071	0.012	0.023
t03o	Text	0.146(+0%)	0.171	0.118	0.477	0.371	0.230	0.068	0.031	0.007
	ApLQA*	**0.210(+44%)**	0.249	0.144	0.562	0.463	0.358	0.106	0.103	0.017
	HpLQA-V100*	0.203(+39%)	0.249	0.144	0.538	0.458	0.328	0.109	0.098	0.014
	HpLQA-V50*	0.204(+40%)	0.247	0.141	0.542	0.459	0.340	0.107	0.099	0.014
	HpLQA-V30*	0.205(+40%)	0.244	0.142	0.536	0.459	0.288	0.107	0.110	0.013
t03e	Text	0.192(+0%)	0.231	0.155	0.564	0.402	0.296	0.103	0.134	0.044
	ApLQA*	0.246(+27%)	0.301	0.179	0.628	0.491	0.383	0.133	0.296	0.049
	HpLQA-V100*	**0.247(+28%)**	0.305	0.184	0.610	0.484	0.366	0.141	0.280	0.049
	HpLQA-V50*	0.243(+26%)	0.293	0.184	0.610	0.484	0.353	0.139	0.282	0.042
	HpLQA-V30*	0.230(+19%)	0.295	0.183	0.609	0.486	0.292	0.140	0.303	0.026
t04o	Text	0.078(+0%)	0.178	0.107	0.361	0.188	0.012	0.033	0.046	0.044
	ApLQA*	**0.110(+41%)**	0.220	0.119	0.381	0.255	0.053	0.044	0.110	0.051
	HpLQA-V100	0.100(+28%)	0.184	0.111	0.379	0.249	0.028	0.043	0.086	0.038
	HpLQA-V50	0.103(+31%)	0.183	0.110	0.381	0.251	0.085	0.039	0.087	0.039
	HpLQA-V30	0.105(+34%)	0.193	0.115	0.378	0.251	0.095	0.039	0.096	0.042
t04e	Text	0.097(+0%)	0.184	0.125	0.461	0.199	0.003	0.041	0.152	0.048
	ApLQA*	**0.135(+38%)**	0.239	0.149	0.471	0.268	0.030	0.043	0.237	0.070
	HpLQA-V100	0.128(+31%)	0.219	0.140	0.472	0.262	0.062	0.043	0.213	0.059
	HpLQA-V50	0.126(+29%)	0.220	0.138	0.474	0.264	0.008	0.041	0.217	0.060
	HpLQA-V30	0.127(+30%)	0.223	0.142	0.473	0.264	0.011	0.041	0.218	0.060
t05o	Text	0.073(+0%)	0.207	0.175	0.339	0.141	0.015	0.097	0.075	0.016
	ApLQA*	0.129(+77%)	0.294	0.212	0.397	0.209	0.042	0.104	0.328	0.017
	HpLQA-V100*	**0.130(+77%)**	0.311	0.213	0.390	0.216	0.039	0.103	0.319	0.017
	HpLQA-V50*	0.130(+77%)	0.300	0.217	0.388	0.219	0.029	0.104	0.322	0.016
	HpLQA-V30*	0.129(+77%)	0.304	0.215	0.386	0.216	0.027	0.103	0.327	0.017
t05e	Text	0.103(+0%)	0.253	0.210	0.389	0.178	0.038	0.095	0.181	0.029
	ApLQA*	0.170(+64%)	0.390	0.266	0.441	0.269	0.084	0.098	0.452	0.030
	HpLQA-V100*	0.170(+64%)	0.390	0.268	0.443	0.282	0.068	0.096	0.434	0.032
	HpLQA-V50*	0.172(+65%)	0.388	0.269	0.439	0.283	0.070	0.100	0.434	0.032
	HpLQA-V30*	**0.172(+66%)**	0.399	0.271	0.435	0.279	0.071	0.097	0.453	0.032

Table 5.8: Comparison of text retrieval, ApLQA and HpLQA.

query classes based on the model selection principle. The experimental settings are similar to those presented above except that the results of $t02o$ are evaluated in addition. Note that, QClass-x in this table are learned on a larger training set and thus it can outperform its counterparts in Table 5.6, which shows the importance of collecting sufficient training data. Despite this change, both ApLQA and KpLQA can still outperform QClass-x by a margin of 1-2% (10-20% relatively) w.r.t. MAP. Similarly, the increase in precision is the main advantage brought by the pLQA models. Among these models, KpLQA shows some advantages over the ApLQA model in 3 out of 4 collections, although the difference between them is not significant. However, we believe KpLQA has more opportunities improved because it is flexible to incorporate the distance-metric-type features and we expect to explore this choice in the future.

Table 5.8 compares text retrieval, ApLQA and HpLQA with three variance settings, i.e., $\sigma^2 = 30$, $\sigma^2 = 50$ and $\sigma^2 = 100$. All the parameters are estimated from the external training set $t04dx$ and each pLQA model is learned with six query classes. In this table, we can observe that sometimes HpLQA can outperform ApLQA by a margin of 1% (6% relatively), however, the improvement is not yet consistent over all the search collections. Especially for the corpora of $t03o$, $t04o$ and $t04e$, HpLQA even generates slightly inferior results compared with ApLQA even if HpLQA can take into account the distribution of combination weights inside a query class. This degradation might be due to the problems from any of the following aspects: 1) assumption of Gaussian noise for combination weights; 2) approximate inference process; 3) lack of sufficient training data to support the model. Therefore, further experiments are still needed to uncover the underlying issues and explore the full potentials of the HpLQA model.

5.4.3 Upper Bounds for Combination

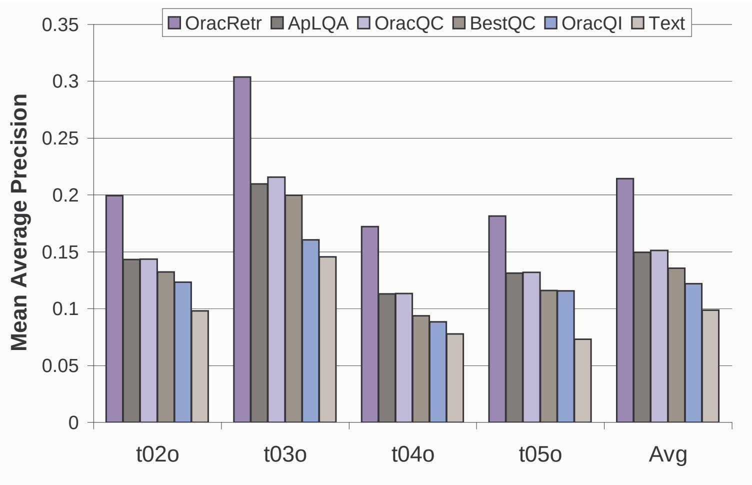


Figure 5.14: Comparison of retrieval models and retrieval upper bounds averaged over TRECVID'02-'05.

As mentioned before, query-independent combination methods are often too limited to support the task of multimedia retrieval and this observation motivates us to develop more advanced query-dependent combination models. Therefore in this section, we would also like to study the retrieval upper bounds for the query-class combination models in order to gain more insights for our task. We plot Figure 5.14 to compare three retrieval models including text retrieval baseline(Text), our best query-class combination model(BestQC), our best ApLQA model(ApLQA), as well as three combination upper bounds, including the global linear upper bound (OracQI), the query-class combination upper bound given that the query classes are pre-defined (OracQC) and the local linear upper bound (OracRetr). Five columns are presented where the first four columns correspond to the results from TREC'02 to TREC'05 and the last column summarizes the average performance. The upper bounds can be computed via the MCS algorithm by assuming the testing ground truth is known. By analyzing this figure, it is interesting to find that the best query-class model learned from development sets can even produce better outputs than the oracle query-independent model directly learned from search collections. Furthermore, the best ApLQA model, which can automatically discover the latent query classes from the development data, can even achieve a close performance to the oracle query-class model in all four collections. This property is intriguing especially when we are dealing with large multimedia corpora. However, given the visible gap between OracRetr and ApLQA, there is still room for us to improve the retrieval models by better exploiting the query information.

5.5 Conclusion

This chapter has described two types of query-dependent retrieval models in an attempt to incorporate the query information into the combination process. The first type is called query-class dependent retrieval model. Its basic idea is to first classify each query into one of the predefined classes and then apply the associated combination weights, which are learned from the development data off-line, to fuse the outputs from multiple retrieval sources. The experimental results demonstrate that applying query-class dependent weights can considerably improve the retrieval performance over using query-independent weights.

In order to automatically detect query classes from development data, we also propose a series of retrieval models called probabilistic latent query analysis (pLQA) to merge multiple retrieval sources, which unifies the combination weight optimization and query class categorization into a discriminative learning framework. Four pLQA models have been discussed which evolve from a basic version(BpLQA) to an adaptive version (ApLQA) that operates on the query feature space, a kernel version (KpLQA) that builds on a Mercer kernel representation, a hierarchical version that bases itself on a hierarchical Bayesian model. In contrast to the typical query-independent and query-class combination methods, pLQA can automatically discover latent query classes from the training data rather than relying on manually defined query classes. Also, it can associate one query with a mixture of query classes and thus non-identical combination weights. Finally, based on statistical model selection principles, we can obtain the optimal number of query classes by maximizing the regularized likelihood. Our experiments in large-scale multimedia retrieval task demonstrate the superiority of the proposed methods which can achieve significant gains in average precision over the query-independent/query-class combination methods. We expect that future investigation on designing better query features for ApLQA and introducing some distance-metric-type kernels to KpLQA could result in a further improvement on the performance of retrieval source combination.

5.6 Discussions

5.6.1 Query Classification and Clustering

Query classification has been widely investigated in the community of information retrieval and query answering. Li et al. [LR02] and VideoQA [YCZ$^+$03] adopt a hierarchial classification approach to categorize free-form factual queries. Five types of machine learning approaches are experimented by Dell et al. [ZL03] for automatic question classification task. It could be found that the definition of our query types is different from the definitions of question categories in the questions answering task(such as searching for location, numeric number and description [LR02]), because the purpose of our task is to improve the combination of multiple video descriptors rather than extracting exact answers from text archives as in question answering. He et al. [HO04] proposed a query-based pre-retrieval strategy, i.e., to select the best retrieval models based on query clusters defined on some intrinsic features such as query length and ambiguity. To improve the web document retrieval, Kang et al. [KK03] classify the user queries into three categories, that is, the topic relevance task, the homepage finding task and the service finding task using various statistics from query words. Different linear weights of text information and hyperlink information will be assigned based on the query categories. To handle the task of distributed information retrieval, Voorhees [VGJL95] proposed a query clustering method for distributed IR, which groups the queries based on the number of common documents retrieved, and create the centroid by averaging the query vectors inside the clusters. For the task of novelty detection, Yang et al. [YZCJ02] classified the documents into several pre-defined topics and performed topic-conditioned novelty detection for documents in each topic. In the domain of image retrieval, Benitez et al. [BBC97] proposed a content-based meta-search image search engine named Metaseek, which assigns the new query images to one of the predefined clusters and selects one of the target image search engines based on their previous success of handling the similar queries. Recently, query difficulty detection [YTFCD05] has emerged as one of the most interesting directions in IR, which attempts to quantify the quality of retrieval results returned by a given system. Such a detection can helpful to several retrieval applications, such as using query difficulty as an indicator to decide the combination weights accordingly for different retrieval results.

In contrast to above query classfication/clustering methods, the query analysis approaches attempt to extract the latent query mixtures not only based on the similarities between queries (or query features), but also based on the similarities of the associated combination parameters. This is because the main goal of query analysis is to predict better combination functions from the past retrieval results rather than grouping queries into several unknown clusters.

5.6.2 Latent Semantic Analysis

There are numerous approaches available for capturing low-dimensional latent semantic representations in the context of information retrieval. Latent semantic indexing (LSI) [DDF$^+$90] finds a linear transform of word counts into a latent eigenspace of document semantics. The probabilistic latent semantic indexing (pLSI) model [Hof99] extends LSI to a probabilistic framework by assuming words to be (marginally) *iid* samples from a document-specific mixture of word distributions. The mixture of unigrams model [BNJ03] is a special case of pLSI where each document is associated with only one topic. The latent Dirichlet allocation (LDA) model by Blei et al. [BNJ03] offers a more expressive and generalizable topic-mixing scheme by associating with each document a unique latent topic-mixing vector represented by a random point in a simplex. Each word is independently sampled according to different topic draws from the topic mixture. In contrast to latent semantic analysis approaches for text modeling, the query analysis approaches have to additionally model the combination of knowledge sources. In some sense, the latent query analysis performs latent semantic analysis on the space of combination parameters, although the parameters are not directly observable to the systems. Note that in our case the dimensionality of observed variables is much smaller than the general setting of text modeling. Therefore, the learning process might require relatively less training examples to achieve an accurate parameter estimation.

Similar ideas have also been applied in the tasks of speaker adaptation and collaborative filtering. For examples, a related approach called EigenVoice [KJNN00] has been proposed for speaker adaptation. It begins with performing principle component analysis(PCA) on the space of hidden markov model(HMM) parameters for the entire set of training speakers. A small number of adaptation data from current speakers are then used to learn a weighted combination of the eigenvetors, which can reasonably approximate the intrinsic parameters of the current speaker with little training data needed. Hofmann [HP99] successfully applied the latent semantic analysis techniques in collaborative filtering under the assumption that the observed user ratings can be modeled by a mixture of user communities and users may participate probabilistically in one or more of those communities.

Chapter 6

Context Analysis

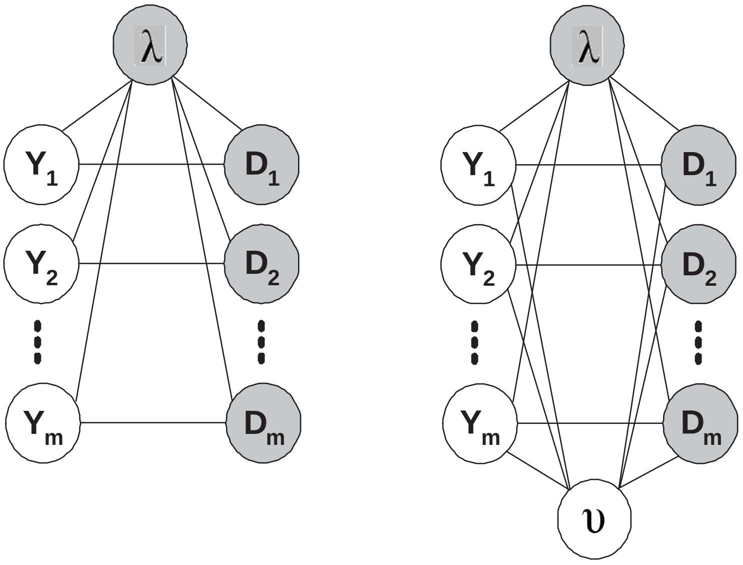
Query analysis offers a useful way to incorporate query information into the knowledge source combination, e.g., learning query independent combination models and query-class based combination models. However, since these learning approaches can only capture general patterns that distinguish relevant and irrelevant training documents, their power is usually limited by the number of available manual relevance judgments. If a high-level semantic concept is either very rare or has an insignificant discriminative pattern in the training data, it will simply be ignored by the learning algorithm without showing any effects in the combination function. In the rest of the paper, we call these *unweighted semantic concepts* in short. In fact, these unweighted semantic concepts constitute a major proportion of the available high-level semantic concepts. For example, even after learning with a large development collection including 500 hours of video from TRECVID'03-'05, a five query-class combination model still ignores more than 80% of the semantic concepts due to their inability to show strong patterns in the training documents. However, many unweighted semantic concepts are not completely worthless and occasionally they are helpful for the queries in related domains. For instance, the infrequent appearance of the concepts "ocean" and "sand" usually result in their absence in the learned retrieval function. But they can become highly predictive if the current query is "finding people on the beach".

In fact, if we do not have any training data at all, the simplest approach to handle high-level semantic concepts is to match the name of each concept with the query terms. If a concept name is found to be relevant, then its detection outputs can be used to refine the initial retrieval results. For example, the semantic concept of "building" detection will be helpful for retrieving the query of "finding the scenes containing buildings in New York City". However, in practice it is unrealistic to expect a general user to explicitly indicate all related concepts in his query description [CH05]. To extend the power of simple query matching, we can follow the idea of global query expansion strategies in text retrieval, which attempts to enrich the query description from external knowledge sources such as a co-occurrence thesaurus [QF93] created based on global term-to-concept similarities, or a semantic network organized to provide semantic relations between keywords, e.g., WordNet [Fel98]. These approaches have been successfully applied in the multimedia retrieval task [VN06, NZKC06]. However, apart from some useful semantic concepts, these approaches are also likely to introduce additional noisy concepts to the query and thus suffer from unexpected deterioration of the search outputs. They are also not able to build up the connections between queries and (hidden) semantic concepts that do not have any explicit semantic meaning. Moreover, even when the subset of relevant concepts are noiseless, it remains a challenge for such kinds of query matching approaches to derive a good strategy to combine all the high-level semantic concepts with other text/image retrieval results.

Following the above discussion, we find that it is more desirable to develop retrieval approaches that can adaptively leverage unweighted semantic concepts on a per query basis without the support of training data. Inspired by the local analysis approaches, we propose a new retrieval approach called probabilistic local context analysis(pLCA), which can automatically leverage useful high-level semantic concepts to improve the initial retrieval output. It can be formally described as an undirected graphical model that treats the document relevances and the combination weights of concepts as a set of latent variables. In this model, the marginal dependence between initial combination weights and latent combination weights allow the usefulness of each unweighted concept to be determined in the retrieval process. We also propose a pLCA variant that can incorporate human relevance feedback into the learning process. Our video retrieval experiments on TREC'03-'05 collections have demonstrated the effectiveness of the proposed pLCA approaches, which can achieve noticeable performance gains over various baseline methods.

6.1 Probabilistic Local Context Analysis

In this section, we present the proposed retrieval model called probabilistic local context analysis(pLCA), followed by describing its parameter estimation and inference approaches as well as its connections to other methods. We also extend



(a) (b)

Figure 6.1: (a) The graphical model representation for the basic probabilistic retrieval model, where the document relevance is only determined by the initial combination weights λ and the corresponding ranking features, (b) The graphical model representation for pLCA, which assumes the weights ν of unweighted semantic concepts to be latent variables. The nodes with known values are shaded, while other nodes are unshaded.

pLCA to a variant that can handle additional human relevance feedback, Finally we conclude with an illustrative example on a two-dimensional synthetic dataset.

6.1.1 Model Description

Let us begin by introducing the basic notations and terminologies used in this work. The term *document* is referred to as the basic unit of retrieval throughout this paper. The current query is denoted as Q where it can have either a set of keywords, a detailed text descriptions or even possibly image, audio, video query examples. A search collection $\mathcal{D}$ contains a set of documents $\{D_1, ..., D_j, ..., D_{M_D}\}$. Let $y_j \in \{-1, 1\}$ indicates if the document D_j is relevant or irrelevant to the query Q. For document D_j, we can generate a bag of ranking features denoted as $f_i(D_j), i = 1..N$. In this paper, we assume the ranking features are generated from the indexing outputs of unweighted semantic video concepts. Moreover, each ranking feature $f_i(D_j)$ is associated with a combination weight ν_i to indicate its influence on the retrieval function. All the bold letters represent a vector of the variables, e.g., $\mathbf{y}$ means $\{y_1, ..., y_{M_D}\}$.

We begin with a review of the basic multimedia retrieval models, of which the underlying idea is to utilize discriminative models to combine multiple retrieval sources. Formally, we model the posterior probability of the relevance as a logistic function on a linear combination of ranking features, i.e.,

$$P(y|D, \lambda) = \sigma\left(y \sum_{i=0}^{N} \lambda_i f_i(D)\right), \tag{6.1}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the standard logistic function and λ is the estimated combination parameter for the output of i^{th} ranking features $f_i(D)$. This logistic regression model, a.k.a. the maximum entropy model, summarizes our basic retrieval source combination framework. Once the parameters are estimated, documents can be presented to users in descending order of $P(y_+|D)$, or equivalently by the weighted sum of retrieval outputs $\sum_{i=0}^{N} \lambda_i f_i(D)$. By summarizing all the document prediction into a vector representation and eliminating the normalization factor, we can have

$$P(\mathbf{y}|D, \lambda) \propto \prod_{j=1}^{M_D} \exp\left(y_j \sum_{i=0}^{N} \lambda_i f_i(D_j)\right). \tag{6.2}$$

This formulation is equivalent to the previous representation due to the independence assumption between document relevances. Its graphical model representation is shown in Figure 6.1(a), where the document relevance is determined by the initial combination weights λ together with the corresponding ranking features $f(D_j)$, and thus the document relevance variables Y_j are independent to each other.

In order to automatically leverage additional semantic concepts, we propose a novel retrieval model called probabilistic local context analysis(pLCA) by considering the combination weights of unweighted semantic concepts as latent variables in an undirected graphical model. In more detail, we assume that we have generated all the ranking features for each document D_j and a set of initial combination parameters λ which can be estimated from a variety of methods such as manual definition and automatic learning. Then the combination weights ν corresponding to the unweighted semantic concepts, i.e., the semantic concepts that are not used to generate the initial outputs (equivalently, their associated λ are set to 0), are left as latent variables. Finally, both λ and ν will impose a joint effect on the document relevance variables Y_j. We expect that the distribution of unknown ν_i can be influenced by the initial retrieval results through Y_j and thus the useful ranking features can be selected without manual intervention.

Based on the model description, we can derive its corresponding graphical model representation as shown in Figure 6.1(b). In analogy to a conditional random field [LMP01], we can derive the conditional probability of relevance $\mathbf{y}$, latent weights ν given initial weights λ and documents $\mathbf{D}$ as follows,

$$p(\mathbf{y}, \nu | \lambda, \mathbf{D}) \propto \prod_{l} p_0(\nu_l) \prod_{j=1}^{M_D} \exp \left(y_j \sum_{i \in W} \lambda_i f_i(D_j) + y_j \sum_{l \in U} \nu_l f_l(D_j) \right). \quad (6.3)$$

where $W = \{i : \lambda_i \neq 0\}$ contains the indices of initially weighted semantic concepts, $U = \{i : \lambda_i = 0\}$ contains the indices of unweighted semantic concepts, ν_l is a latent combination weight for the l^{th} unweighted concepts[1]. The prior distribution $p_0(\nu_l)$ represents how likely it is that an unweighted semantic concept is relevant to the user's information need only based on the query description. For example, the query of "finding the map of Iraq" can induce a high weight on the semantic concept of "map". However, this prior term is not able to capture the semantic concepts that are not explicitly mentioned in the query. Therefore, to further refine the document relevances, we use the potential $\exp(y_j \lambda_i f_i(D_j))$ to capture the effects from initial combination parameters and the potential $\exp(y_j \nu_l f_l(D_j))$ to model the connections from additional unweighted concepts. Since both λ_i and $f_i(D_j)$ are already given, we can pre-compute the initial retrieval results $f^{\lambda}(D_j) = \sum_i \lambda_i f_i(D_j)$ and simplify the Eqn(6.3) to be,

$$p(\mathbf{y}, \nu | \lambda, \mathbf{D}) \propto \prod_{l} p_0(\nu_l) \prod_{j=1}^{M_D} \exp \left(y_j f^{\lambda}(D_j) + y_j \sum_{l \in U} \nu_l f_l(D_j) \right). \quad (6.4)$$

The construction under undirected graphical model semantics is of crucial importance for the correct functionality of pLCA. The conditional dependence between λ and ν allows the posterior probability of $p(\nu|\lambda)$ to be updated according to the initial parameters. Therefore, the proposed pLCA model is able to estimate the effectiveness for each concept and discover useful semantic concepts from the query context. In contrast, if we switch the model representation to be a directed

[1] We use the proportional sign to indicate the intractability to compute the normalization constant on the right hand side. The same as follows.

graph, the latent combination weight ν will be then independent to the initial ranking results λ given y_j is unknown. Since no additional information from λ can flow to the nodes of ν, a directed graphical model will trivially produce the same retrieval outputs as the original outputs.

The potential functions can be chosen in a flexible way. For example, we can model the prior potential to be a normal distribution $\mathcal{N}(\nu_l^0, \sigma^2)$ where ν_l^0 reflects our prior belief on the combination weight for l^{th} ranking feature and σ^2 is a pre-defined constant variance. $\nu_l^0 > 0$ if the concept is mentioned in the query, otherwise $\nu_l^0 = 0$. Furthermore, if we choose the potential between ν, y and D to be a sigmoid function $\tanh(x) = 1/(1 + \exp(-x))$, we can rewrite Eqn(6.4) as,

$$p(\mathbf{y}, \nu|\lambda, \mathbf{D}) \propto \prod_l p_0(\nu_l) \prod_{j=1}^{M_D} \exp(y_j f^\lambda(D_j)) \tanh(y_j \sum_l \nu_l f_l(D_j)).$$

These choices of potential functions are all sensible and thus both of them are evaluated in our experiments. For the sake of efficiency, we select only the top M_D documents in the initial ranking to update in Eqn(6.3), where M_D is a much smaller number than the number of documents in the entire collection. Their parameter estimation and inference methods are discussed in the following section.

6.1.2 Inference

By marginalizing out the latent variables ν, we can present the documents to users in a descending order of the following conditional probability of y,

$$p(\mathbf{y}|\lambda, \mathbf{D}) = \int_\nu \prod_l p_0(\nu_l) \prod_{j=1}^{M_D} \exp\left(y_j f^\lambda(D_j) + y_j \sum_{l \in U} \nu_l f_l(D_j)\right) d\nu. \qquad (6.5)$$

However, because of the presence of the normalization constant on the right hand side, it is usually intractable to compute the posterior probability in Eqn(6.5) with an exact inference approach. Therefore, we resort to variational methods to provide an approximate inference for the intractable posterior distributions. Specifically, we adopt the mean field approximation [PA87] in our derivation, which takes a factorized form of all singleton marginals over the variables.

The first step of the mean field approximation is to construct the following family of variational distributions,

$$q(\mathbf{y}, \nu) = \prod_j q(\nu_l|\beta_l) \prod_j q(y_j|\gamma_j),$$

as a surrogate to approximate the posterior distribution $p(\mathbf{y}, \nu|\mathbf{a}, \mathbf{D})$, where $q(\nu_l|\beta_l)$ is a Gaussian distribution with mean β_l and the same variance σ as the prior potential $p_0(\nu)$, $q(y_j|\gamma_j)$ is a Bernoulli distribution where $y_i = 1$ with a sample probability of γ_j and otherwise $y_i = -1$. The independence between variables in the variational distributions results in an efficient inference algorithm as shown below. We aim to optimize the KL divergence between $q(\mathbf{y}, \nu)$ and $p(\mathbf{y}, \nu|\lambda, \mathbf{D})$. This optimization can alternatively cast the maximization of the following lower bound,

$$\begin{aligned} 0 &\geq KL(q(\mathbf{y}, \nu)||p(\mathbf{y}, \nu|\lambda, \mathbf{D})) \\ &= E_q\left[\log p(\mathbf{y}, \nu|\lambda, \mathbf{D})\right] - E_q\left[\log q(\mathbf{y}, \nu)\right] \\ &= E_q\left[\sum_l \log p_0(\nu_l) + \sum_j y_j(f_j^\lambda + \sum_l \nu_l f_{jl})\right] + H(q) \\ &= -\frac{(\beta_l - \nu_l^0)^2}{2\sigma_l^2} + \sum_j (2\gamma_j - 1)(f_j^\lambda + \sum_l \beta_l f_{jl}) \\ &\quad + \sum_j \gamma_j \log \gamma_j + \sum_j (1 - \gamma_j) \log(1 - \gamma_j), \end{aligned} \tag{6.6}$$

where f_{jl} denotes $f_l(D_j)$, f_j^λ denotes $f^\lambda(D_j)$, $E_q[f]$ refers to the expectation of $f(x)$ with respect to the distribution of $q(x)$, $H(q)$ refers to the entropy of the distribution q. It could be found that the gap between the inequality is exactly the K-L divergence between the variational posterior distribution $q(\mathbf{y}, \nu)$ and true posterior distribution $p(\mathbf{y}, \nu|\lambda, \mathbf{D})$. Therefore, we can alternatively solve a simpler optimization problem, i.e., to maximize the variational lower bound in order to find the best variational distributions to approximate the true posterior distribution.

By taking the derivative of the variational low bound with respect to each variational parameter to be zero, we can derive the following fixed point equations,

$$\begin{aligned} \gamma_j &= \left[1 + \exp\left(2f_j^\lambda + 2\sum_l \beta_l f_{jl}\right)\right]^{-1}, \\ \beta_l &= \nu_l^0 + \sum_j (2\gamma_j - 1)\sigma_l^2 f_{jl}. \end{aligned} \tag{6.7}$$

These fixed point equations can be interpreted as follows: 1) the first equation attempts to assign a relevance judgment for each document where the top-ranked examples are judged as positive and bottom-ranked examples are judged as negative, 2) the second equation aims to estimate the usefulness of each semantic concepts given the previous judgments. These equations are invoked iteratively until the change of KL-divergence is small enough. Upon convergence, we use the final $q(y_j|\gamma_j)$ as a surrogate to approximate the posterior probability $p(y_j|\lambda, \mathbf{D})$ without explicitly computing the integral. Since $q(y_j|\gamma_j)$ is a Bernoulli distribution, we can simply rank the documents in an descending order of the parameter γ_j as the retrieval outputs. Note that this iterative update process typically converges in a small number of iterations and thus the proposed pLCA approach can be implemented efficiently in a real retrieval system.

If the potential function of ν, y and D is chosen to be a sigmoid function, the fixed point equations should be modified accordingly. In this case, if we further assume the variance of the variational distribution $q(\nu_l|\beta_l)$ to be 0, we can have the updated fixed point equations,

$$\begin{aligned}
\gamma_j &= \left[1 + \exp\left(2f_j^\lambda - 2\sum_l \log(1 + \exp(\sum_l \beta_l f_{jl}))\right)\right]^{-1}, \\
\beta_l &= \arg\min_{\beta_l} \frac{(\beta_l - \nu_l^0)^2}{2\sigma_l^2} + \sum_j (2\gamma_j - 1)\log(1 + e^{\sum_l \beta_l f_{jl}}),
\end{aligned}$$

where the second equation is modified to optimize a regularized logistic regression problem. Similarly, these fixed point equations should be iterated until convergence.

Remark: The update process of pLCA shares some characteristics similar to the traditional pseudo-relevance feedback(PRF) techniques in the sense that both of them aim to refine the retrieval outputs based on initial rankings. However unlike PRF, pLCA does not require the assumption that most of top-ranked documents have to be relevant. Instead, it can work reasonably well as long as the top-ranked documents contain more relevant documents than the bottom-ranked documents. pLCA also provides a sound probabilistic interpretation and a convergence guarantee on the iterative parameter updating process, which is usually missing in the PRF approaches. Moreover, it is not necessary for pLCA to specify a certain number of positive documents for the refinement process, since the initial prediction confidence has been naturally integrated in the probabilistic model.

When using text retrieval outputs as initial search results, pLCA also has connections with the global analysis methods that aim to incorporate representative semantic concepts based on global term-to-concept similarities, because in this case pLCA will first place the documents containing query keywords at the top positions and then select their associated semantic concepts for following updates. For instance, if we set the $2\gamma_j - 1$ to the outputs from a vector space retrieval model

$\sum_w Q_w^{tf} D_{wj}^{tf}$ and $\sigma_l = 1$, then we can rewrite the second equation of Eqn(6.7) to be,

$$\beta_l = \nu_l^0 + \sum_j \sum_w Q_w^{tf} D_{wj}^{tf} f_{jl} = \nu_l^0 + \sum_w Q_w^{tf} \left[\sum_j D_{wj}^{tf} f_{jl} \right],$$

where $\sum_j D_{wj}^{tf} f_{jl}$ can be viewed as the term-to-concept similarity constructed based on the entire collection. But in contrast to global analysis, pLCA does not need to maintain a large mapping table between query terms and semantic concepts. Moreover, pLCA can be initialized from more advanced retrieval methods than text retrieval, such as query-class based combination approach, which is not straightforward to implement in a global analysis method.

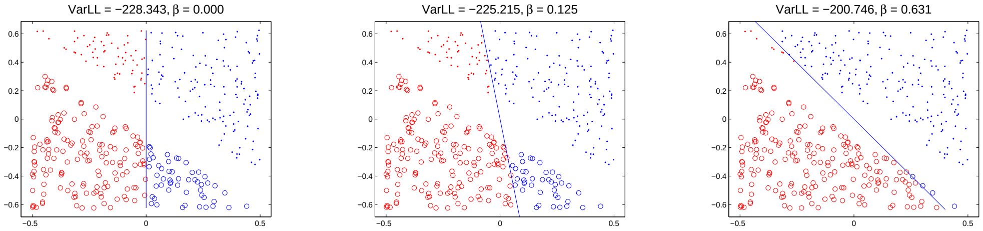
6.1.3 Incorporating Human Feedback

The aforementioned pLCA approach can automatically leverage useful semantic concepts in an unsupervised manner. But in order to further improve the retrieval performance, pLCA can be augmented by incorporating additional human relevance feedback. Typically, relevance feedback algorithms proceed by first requesting users to annotate a small number of video documents from the initial retrieval results and then feeding them back to update the retrieval models. It can be viewed as a learning component in a retrieval system, which learns from a small amount of relevant examples to adjust the ranking function adaptively for additional annotations[2].

In this section, we mainly discuss how to use the additional annotation to modify the combination parameters ν in the pLCA model. Formally, we can denote the manual relevance judgment as $\{y_1, ..., y_K\}(y_k \in \{-1, 1\})$ associated with a set of documents $\{D_1, ..., D_K\}$. Given that a small number of documents have been annotated by the users, we can obtain a similar set of fixed point equations as before except that the variational parameters γ_j do not need to be updated any more on those annotated documents. Therefore, we can have

$$\begin{aligned} \gamma_j &= \left[1 + \exp\left(2f_j^\lambda + 2\sum_l \beta_l f_{jl} \right) \right]^{-1}, \\ \beta_l &= \nu_l^0 + \sum_j (2\gamma_j - 1)\sigma_l^2 f_{jl} + \sum_k y_j \sigma_l^2 f_{lk}. \end{aligned} \tag{6.8}$$

[2] Note that some of relevance feedback algorithms also consider reformulating the text queries or image queries based on the feedback information, however, these types of algorithms are not our major focus.



(a) Iteration 0 (b) Iteration 1 (c) After convergence

Figure 6.2: An illustrative example of pLCA on a 2-D synthetic data set. X-axis stands for the initial retrieval outputs $f^\lambda(D_j)$ and Y-axis stands for the value of an un-weighted ranking feature $f(D_j)$. "·" and "○" denote the relevant and irrelevant documents. Red/blue colors denote the positive/negative predictions from pLCA. (a) The synthetic dataset with the initial decision boundary shown as the solid line. Above the figure, we also show the values of the variational lower bound $VarLL$ and the variational combination parameters β, (b) The decision boundary after one iteration of pLCA, (c) The decision boundary after its convergence.

If we ignore all the variational decision γ_j and only consider the relevance judgment on the feedback documents, the update rules will degrade to a Rocchio-like updating process,

$$\beta_l \quad = \quad \nu_l^0 + \sum_k y_j \sigma_l^2 f_{lk} = \nu_l^0 + \sum_{y_k=1} \sigma_l^2 f_{lk} - \sum_{y_k=-1} \sigma_l^2 f_{lk}. \qquad (6.9)$$

Theoretically, we should update γ_j for each document D_j that is not judged by users. But for the sake of efficiency, we only consider updating γ_j of the top M_D documents that do not associate with any feedback process.

6.1.4 Illustrative Examples

To show the ability of pLCA to automatically discover useful semantic concepts and update initial search results, we prepared a synthetic dataset shown in Figure 6.2(a) where X-axis means the initial retrieval outputs $f^\lambda(D_j)$ and Y-axis means the detection value of an unweighted semantic concept $f(D_j)$. In this figure, "·" and "○" represent the relevant and irrelevant documents. Red and blue colors represent the positive and negative predictions from pLCA. There are a total of 200 relevant documents and 200 irrelevant documents. The prediction for each

Data Set	**t03o/t03e**	**t04o/t04e**	**t05o/t05e**
Query Num	25	24	24
Doc Num	75850	48818	77979

Table 6.1: Labels of video collections and their statistics. $t**o/e$ indicate search sets with original or expanded query keywords.

document D_j is determined by its corresponding variational parameter γ_j with a threshold of 0.5, or equivalently, whether $f^\lambda(D_j) + \beta f(D_j)$ is larger than 0. Figure 6.2(a) also uses a solid line to indicate the initial decision boundary, which is purely determined by $f^\lambda(D_j)$ at the beginning. From this graph, we can observe that neither the initial prediction $f^\lambda(D_j)$ or the ranking feature $f(D_j)$ is a perfect predictor for the document relevances, but both of them can provide informative evidence to indicate the ground truth. More importantly, since the initial retrieval results provide a better-than-random performance, it can serve as a starting point to detect the usefulness of the unweighted concept.

Figure 6.2(b) plots the decision boundary after running one step of the fixed point equations in Eqn(6.7). It can be found that the variational combination parameter β becomes a positive number and the decision boundary is shifted to a more accurate position. After all the fixed point equations run to converge, Figure 6.2(c) shows the final decision boundary which produces a much better retrieval results than its initial setting. Note that, though the final decision is also related to the prior variance defined in Eqn(6.7), the retrieval performance in general is insensitive to the change of this parameter as demonstrated in our following experiments.

6.2 Experiments

Our experiments are designed based on the guidelines of the manual retrieval task in the TREC video retrieval evaluation(TRECVID) [SO03], which requires an automatic video retrieval system to search relevant documents without any human feedback. The retrieval units were video shots defined by a common shot boundary reference. The query topics contain multimodal information including text descriptions, image examples and video examples. We used the query topics and video collections from TREC'03-'05 to evaluate the proposed learning algorithms. Each of these video collections is split into a development set and a search set chronologically by source. The development sets are used as the training pool to develop automatic multimedia retrieval algorithms and the search sets mainly serve as the testbeds for evaluating the performances of retrieval systems. All of our experiments are evaluated on the search sets where for each query topic, the relevance judgment on search sets was provided officially by NIST. The development sets are only used to build the models for semantic concepts and learn the combination function in baseline methods. The computation of pLCA has no relations to the development sets at all. Table 6.1 lists the labels of each search collection and their statistics of query/document numbers.

As the building blocks of video retrieval, we generated a number of ranking features on each video document including 75 high-level semantic concepts learned from development data (including face, anchor, commercial, studio, graphics, weather, sports, outdoor, person, crowd, road, car, building, motion and so forth), and 5 uni-modal retrieval experts (text retrieval, face recognition, image-based retrieval based on color, texture and edge histograms). The detailed descriptions on the feature generation can be found in [HCC$^+$04]. To avoid the problems brought by inconsistent scales of various retrieval outputs, we transformed the raw scores into their ranks in each ranking feature and normalized them into the range of [0,1] where the highest ranked document corresponds to one and the lowest ranked document is zero. Moreover, as suggested in our previous work [YH06a] that attempts to incorporate ranking information in the learning process, we placed a stronger weight on positive data to balance the positive/negative data distribution and meanwhile shifted the median value of each feature to zero. Please refer to [YH06a] for more details of this rank-based adjustment.

In order to improve the robustness of the pLCA algorithm, we apply a χ^2 test [YP97] to filter out some irrelevant ranking features before the learning process. The χ^2 statistics are generally computed to measure the dependence between two random variables. In our work, we use it to measure the independence between each feature and document relevance. If a feature tends to be independent of the relevance labels, this feature will be eliminated in the training process. Only those features with a strong indication of their dependence are maintained in the learning model. Under the assumption that irrelevant features are less likely to be strongly correlated with the relevance labels, the χ^2 test is able to eliminate most of the irrelevant features and improve the learning robustness, although a small proportion of relevant features might also be mistakenly discarded. In our experiments, we set the cutoff threshold to be 5.02 with a confidence interval of 2.5% in the χ^2 distribution. Setting a higher threshold in the χ^2 test can eliminate more irrelevant features and thus suffer less from the noisy labeling problem, but the

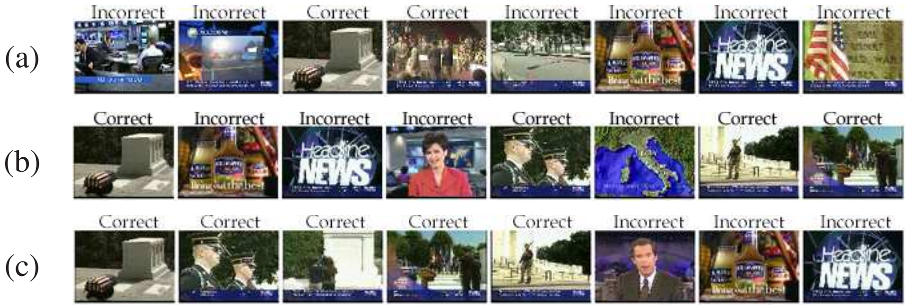


Figure 6.3: The key frames of top 8 retrieved shots for query "Finding Tomb at Arlington National Cemetery" from (a) retrieval on text features, (b) pLCA without image examples and (c) pLCA with image examples.

Query	**Useful concepts and their weights**
Hu Jintao	Leader:1.0, Crowd:0.39, Airplane:0.07
Tony Blair	Leader:1.0, Commercial:-0.37, Crowd:0.93
Helicopter	Airplane:0.23, Sky:1.0
Fire/flame	Car:0.51, Building:0.64, Urban:0.93
Basketball	Crowd:0.53, Commercial:-0.35
Map of Iraq	Maps:1.00, Computer Screen:0.13

Table 6.2: Examples of useful semantic concepts and their corresponding combination weights β found by pLCA on six TRECVID'05 query topics.

performance improvement might be smaller. But because the choice of threshold is relatively insensitive to the retrieval results, we do not show any experiments in this paper on varying the χ^2 threshold.

6.2.1 Retrieval Results

Figure 6.3 compares the performance between pLCA with/without images examples and text retrieval. As we can see, considerable improvement has been achieved by pLCA because more relevant documents are moved to the top. This is achieved by the successful reranking of initial retrieved video shots. Moreover, we can find that better re-ranking results can be obtained by incorporating additional image retrieval experts. To illustrate the ability of pLCA to automatically leverage useful semantic concepts, Table 6.2 lists the examples of six TREC'05 query

Data	Initial	pLCA	MAP	P30	P100	Person	SObj	GObj	Sport	Other
t03o	Text	None	0.146(+0%)	0.171	0.118	0.371	0.230	0.068	0.031	0.007
		PRF	0.149(+2%)	0.181	0.120	0.327	0.286	0.067	0.036	0.009
		Exp	0.170(+16%)	0.197	0.125	0.399	0.321	0.067	0.035	0.008
		Log	0.168(+15%)	0.192	0.124	0.403	0.311	0.066	0.034	0.008
	QClass	None	0.200(+0%)	0.236	0.137	0.466	0.336	0.088	0.106	0.015
		PRF	0.195(-2%)	0.232	0.137	0.441	0.333	0.088	0.104	0.017
		Exp	0.204(+2%)	0.231	0.136	0.472	0.354	0.087	0.103	0.015
		Log	0.203(+1%)	0.228	0.132	0.478	0.342	0.086	0.099	0.015
	ApLQA	None	0.210(+0%)	0.249	0.144	0.463	0.358	0.106	0.103	0.017
		PRF	0.202(-3%)	0.252	0.144	0.434	0.354	0.102	0.106	0.017
		Exp	0.230(+9%)	0.260	0.142	0.480	0.443	0.106	0.108	0.016
		Log	0.222(+5%)	0.257	0.143	0.473	0.410	0.106	0.104	0.017
t04o	Text	None	0.078(+0%)	0.178	0.107	0.188	0.012	0.033	0.046	0.044
		PRF	0.073(-5%)	0.175	0.105	0.160	0.007	0.037	0.042	0.053
		Exp	0.083(+6%)	0.184	0.104	0.189	0.006	0.035	0.072	0.047
		Log	0.090(+15%)	0.210	0.109	0.196	0.007	0.037	0.082	0.058
	QClass	None	0.094(+0%)	0.199	0.125	0.194	0.080	0.046	0.108	0.045
		PRF	0.094(+0%)	0.180	0.125	0.198	0.080	0.050	0.098	0.044
		Exp	0.102(+9%)	0.207	0.120	0.211	0.058	0.047	0.120	0.055
		Log	0.104(+10%)	0.214	0.120	0.215	0.089	0.048	0.115	0.053
	ApLQA	None	0.110(+0%)	0.220	0.119	0.255	0.053	0.044	0.110	0.051
		PRF	0.097(-12%)	0.187	0.119	0.219	0.030	0.045	0.099	0.044
		Exp	0.114(+4%)	0.238	0.124	0.258	0.056	0.041	0.130	0.056
		Log	0.117(+6%)	0.235	0.124	0.267	0.056	0.046	0.128	0.054
t05o	Text	None	0.073(+0%)	0.207	0.175	0.141	0.015	0.097	0.075	0.016
		PRF	0.079(+7%)	0.232	0.184	0.153	0.019	0.102	0.082	0.016
		Exp	0.080(+10%)	0.231	0.185	0.154	0.026	0.104	0.081	0.016
		Log	0.081(+10%)	0.236	0.187	0.159	0.022	0.101	0.082	0.016
	QClass	None	0.116(+0%)	0.292	0.211	0.173	0.031	0.100	0.322	0.017
		PRF	0.111(-4%)	0.281	0.211	0.169	0.031	0.094	0.294	0.018
		Exp	0.124(+7%)	0.304	0.223	0.193	0.035	0.100	0.337	0.017
		Log	0.121(+3%)	0.283	0.219	0.183	0.030	0.102	0.332	0.017
	ApLQA	None	0.129(+0%)	0.294	0.212	0.209	0.042	0.104	0.328	0.017
		PRF	0.125(-3%)	0.294	0.212	0.210	0.048	0.103	0.283	0.017
		Exp	0.137(+6%)	0.315	0.220	0.225	0.044	0.116	0.335	0.017
		Log	0.137(+6%)	0.306	0.220	0.223	0.046	0.115	0.342	0.017

Table 6.3: Comparison of three baseline approaches, pseudo-relevance feedback and their pLCA-augmented retrieval outputs.

topics (in the first column) together with their corresponding semantic concepts and variational combination weights β found by pLCA (in the second column). The query-class based combination method [YYH04] is used to provide the initial search results for pLCA. For each query, we normalized the highest combination weight to be 1 and discarded the semantic concepts when the absolute values of their combination weights are less than 0.05. It can be observed that most of the semantic concepts suggested by pLCA are reasonable and closely related to the query topics. For example, for the query of “Hu Jintao”, it is easy to understand that the results can be augmented by using the concepts of “Government Leader” and “Crowd”. The appearance of “Airplane” can be explained by the fact that the

Data	Initial	pLCA	MAP	P30	P100	Person	SObj	GObj	Sport	Other
t03e	Text	None	0.192(+0%)	0.231	0.155	0.402	0.296	0.103	0.134	0.044
		PRF	0.191(0%)	0.236	0.154	0.354	0.340	0.097	0.167	0.039
		Exp	0.213(+11%)	0.261	0.162	0.430	0.367	0.097	0.216	0.028
		Log	0.216(+12%)	0.253	0.164	0.431	0.363	0.097	0.218	0.042
	QClass	None	0.222(+0%)	0.289	0.187	0.409	0.345	0.127	0.293	0.036
		PRF	0.214(-3%)	0.283	0.187	0.383	0.338	0.125	0.288	0.036
		Exp	0.220(0%)	0.284	0.186	0.415	0.343	0.127	0.282	0.026
		Log	0.222(+0%)	0.292	0.189	0.414	0.346	0.128	0.288	0.032
	ApLQA	None	0.251(+0%)	0.301	0.179	0.491	0.383	0.133	0.296	0.049
		PRF	0.231(-7%)	0.288	0.179	0.430	0.354	0.127	0.301	0.050
		Exp	0.262(+4%)	0.295	0.180	0.496	0.451	0.133	0.300	0.031
		Log	0.259(+3%)	0.300	0.180	0.495	0.423	0.131	0.307	0.044
t04e	Text	None	0.097(+0%)	0.184	0.125	0.199	0.003	0.041	0.152	0.048
		PRF	0.092(-5%)	0.174	0.127	0.169	0.004	0.041	0.144	0.059
		Exp	0.108(+11%)	0.226	0.129	0.193	0.004	0.040	0.210	0.066
		Log	0.107(+10%)	0.219	0.128	0.204	0.004	0.040	0.184	0.064
	QClass	None	0.113(+0%)	0.212	0.152	0.206	0.005	0.046	0.233	0.056
		PRF	0.115(+1%)	0.216	0.152	0.208	0.005	0.049	0.240	0.054
		Exp	0.124(+9%)	0.228	0.153	0.226	0.005	0.047	0.248	0.066
		Log	0.123(+8%)	0.233	0.156	0.228	0.005	0.047	0.252	0.060
	ApLQA	None	0.135(+0%)	0.239	0.149	0.268	0.030	0.043	0.237	0.070
		PRF	0.126(-6%)	0.226	0.149	0.231	0.030	0.046	0.242	0.068
		Exp	0.146(+8%)	0.261	0.155	0.272	0.059	0.044	0.257	0.090
		Log	0.144(+6%)	0.254	0.157	0.279	0.042	0.044	0.257	0.079
t05e	Text	None	0.103(+0%)	0.253	0.210	0.178	0.038	0.095	0.181	0.029
		PRF	0.111(+7%)	0.279	0.225	0.195	0.034	0.101	0.196	0.031
		Exp	0.121(+16%)	0.310	0.234	0.219	0.059	0.100	0.196	0.028
		Log	0.119(+14%)	0.304	0.238	0.209	0.062	0.102	0.196	0.029
	QClass	None	0.167(+0%)	0.376	0.265	0.258	0.079	0.096	0.452	0.033
		PRF	0.162(-3%)	0.367	0.265	0.252	0.081	0.093	0.429	0.033
		Exp	0.171(+2%)	0.387	0.265	0.259	0.086	0.104	0.468	0.031
		Log	0.172(+2%)	0.394	0.267	0.262	0.087	0.097	0.475	0.032
	ApLQA	None	0.170(+0%)	0.390	0.266	0.269	0.084	0.098	0.452	0.030
		PRF	0.171(+0%)	0.385	0.266	0.271	0.088	0.097	0.452	0.030
		Exp	0.178(+4%)	0.394	0.271	0.274	0.091	0.115	0.470	0.029
		Log	0.177(+3%)	0.387	0.271	0.274	0.090	0.111	0.468	0.029

Table 6.4: Comparison of three baseline approaches, pseudo-relevance feedback and their pLCA-augmented retrieval outputs.

truth video clips often contain arrival/departure scenes of Hu Jintao in the airport. These kinds of useful concepts are very difficult to be found by merely analyzing the query description. Note that, the learned combination weights can be either positive or negative. For instance, the concept of "Commercial" is assigned with a negative weight for the query of "Basketball", which means that the basketball scenes usually do not contain any commercials.

Next, we present the retrieval performance of pLCA as well as three baseline approaches, i.e., text retrieval, query-class based combination [YYH04] and adaptive probabilistic latent query analysis (ApLQA) [YH06b][3]. The initial retrieval

[3] We do not compare any pseudo-relevance feedback(PRF) approaches because they are mainly

Data	Initial	Var.	MAP	P30	P100	Person	SObj	GObj	Sport	Other
t03o	Text	1	0.170	0.197	0.125	0.399	0.321	0.067	0.035	0.008
		0.1	0.170	0.197	0.125	0.399	0.322	0.067	0.035	0.008
		10	0.168	0.191	0.121	0.401	0.316	0.064	0.034	0.009
	QClass	1	0.204	0.231	0.136	0.472	0.354	0.087	0.103	0.015
		0.1	0.205	0.235	0.137	0.472	0.354	0.087	0.105	0.015
		10	0.186	0.212	0.125	0.424	0.314	0.086	0.103	0.014
	ApLQA	1	0.230	0.260	0.142	0.480	0.443	0.106	0.108	0.016
		0.1	0.230	0.259	0.142	0.479	0.443	0.106	0.108	0.016
		10	0.211	0.237	0.133	0.424	0.409	0.103	0.106	0.015
t04o	Text	1	0.083	0.184	0.104	0.189	0.006	0.035	0.072	0.047
		0.1	0.083	0.183	0.104	0.188	0.006	0.036	0.072	0.048
		10	0.085	0.203	0.105	0.196	0.006	0.031	0.075	0.051
	QClass	1	0.102	0.207	0.120	0.211	0.058	0.047	0.120	0.055
		0.1	0.103	0.207	0.120	0.212	0.058	0.047	0.120	0.055
		10	0.098	0.209	0.126	0.204	0.062	0.043	0.113	0.052
	ApLQA	1	0.114	0.238	0.124	0.258	0.056	0.041	0.130	0.056
		0.1	0.114	0.238	0.124	0.259	0.056	0.041	0.131	0.055
		10	0.113	0.238	0.125	0.250	0.053	0.041	0.128	0.058
t05o	Text	1	0.080	0.231	0.185	0.154	0.026	0.104	0.081	0.016
		0.1	0.081	0.228	0.184	0.155	0.026	0.105	0.081	0.016
		10	0.080	0.225	0.187	0.152	0.026	0.107	0.081	0.016
	QClass	1	0.124	0.304	0.223	0.193	0.035	0.100	0.337	0.017
		0.1	0.125	0.315	0.223	0.192	0.035	0.103	0.340	0.017
		10	0.109	0.271	0.198	0.162	0.029	0.094	0.299	0.017
	ApLQA	1	0.137	0.315	0.220	0.225	0.044	0.116	0.335	0.017
		0.1	0.138	0.312	0.220	0.225	0.047	0.117	0.337	0.017
		10	0.128	0.306	0.216	0.222	0.036	0.112	0.282	0.017

Table 6.5: Comparison of variances σ^2 in the prior potential.

outputs of pLCA is provided by the baseline method that we are comparing with. We also compare pLCA with a pseudo-relevance feedback algorithm, which assumes a subset of the top-rank examples to be positive and updates the combination parameters via the second equation in Eqn(6.3). The number of feedback documents is chosen as the best configuration ranged from 50 to 500 at a step of 50 in the search collection (so we give an unfair advantage to this PRF algorithm). To determine the parameter ν_l^0 in Eqn(6.3), we directly match query terms with the name of semantic concepts. If there is a match between them, we set the corresponding ν_l^0 to be 1 in our experiments. This information is incorporated in both the baseline methods and the pLCA method in order to provide a fair comparison. Among all 75 semantic concepts, we only consider leveraging the unweighted semantic concepts that do not appear in the baseline retrieval functions. For example, we can utilize all 75 concepts to improve text retrieval but just 65 concepts to improve ApLQA. M_D is set to 300 by default which means only

designed for query expansion in text retrieval and to our best knowledge no similar ideas have been applied in this task before.

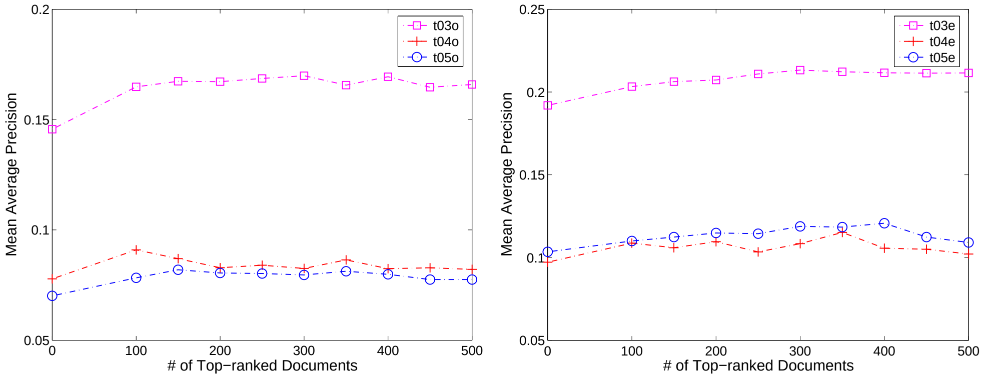
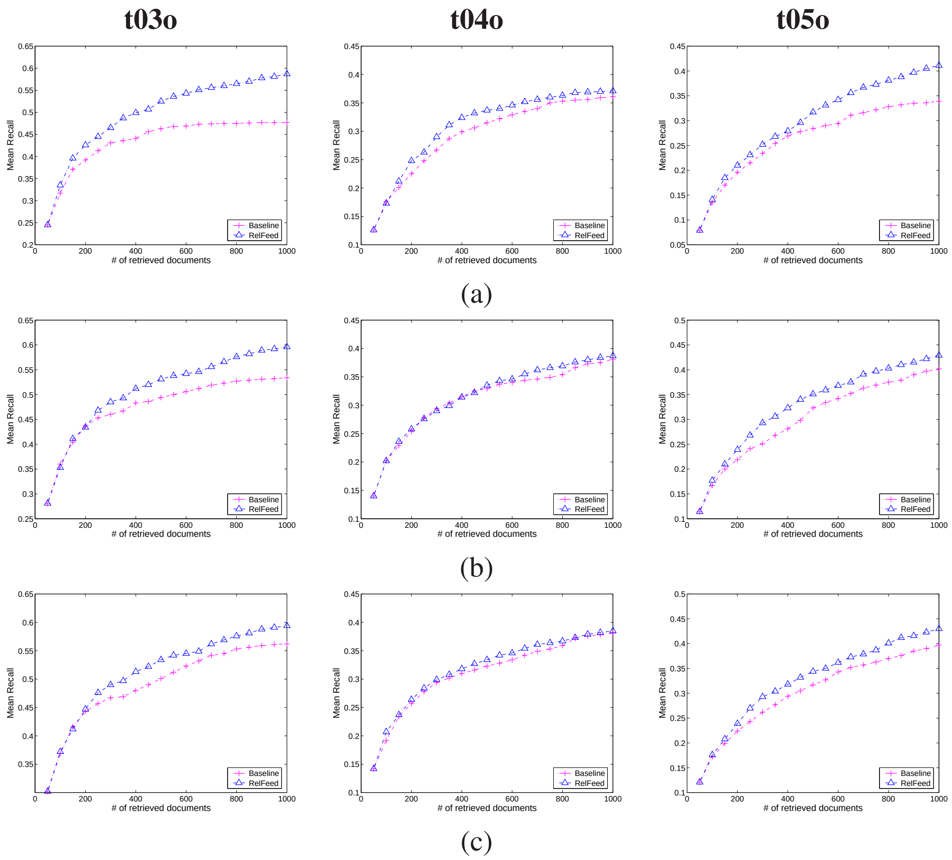


Figure 6.4: Learning curves vs. the number of updated documents in pLCA.

the top 300 documents in the baseline ranking are updated and other documents are left unchanged. The prior variance σ^2 is set to be 1.

Table 6.3 and Table 6.4 provide a detailed comparison between three baseline approaches and their pLCA-augmented outputs on TREC'03-'05. For each baseline method, we apply pLCA with two potential functions $\psi(\cdot)$ including an exponential function(Exp) and a logistic function(Log). All the retrieval results are reported in terms of the mean average precision(MAP) up to 1000 documents and precision at top 30, 100 documents. In this table, we can observe that both pLCA-Exp and pLCA-Log are always superior to the baseline methods no matter which initial retrieval output and data collection are used. On average, it provides a roughly 1-2% absolute improvement in terms of mean average precision (or equivalently 10% relative improvement) over the baseline performance. The major performance growth factor for pLCA can be traced to a higher precision on the top-ranked documents, since pLCA does not aim to increase the recalls. The mean average precision between the settings of pLCA-Exp and pLCA-Log are often comparable to each other with a difference less than 2%. To analyze the results in more detail, we also grouped the queries in each collection and reported their MAP in five different categories, i.e., named person, special object, general object, sports and general queries. By comparing the retrieval performance with respect to each query type, we find that pLCA benefits most from the person-type and special-object-type queries, as well as the sport-type queries in $t05o$. This is because these query types have their information needs clearly defined and thus they are able to be improved by making better use of additional semantic concepts.

To evaluate the sensitivity of pLCA with respect to its parameters, we designed a series of experiments as follows. All of them use text retrieval as initial outputs.



(a)

(b)

(c)

Figure 6.5: The pLCA retrieval performance with additional human feedback. Three baseline approaches are used as the initial search outputs: (a) text retrieval, (b) query-class combination, (c) ApLQA.

Table 6.5 compares pLCA with the prior variance σ^2 varied from 0.1 to 10. As we can observe, the setting of $\sigma^2 = 0.1$ is on par with the setting of $\sigma^2 = 1$. However, if we modify σ^2 to be a large value, e.g., 10 in our case, it might occasionally result in a large loss in terms of mean average precision. This suggests that pLCA works more stable with a smaller variance in its prior potential, because large variances might dilute the useful information encoded in the prior potential. Figure 6.4 depicts the learning curve of pLCA with the number of updated documents M_D grown from 0 to 500 at a step of 50. In these three collections, although the highest mean average precision are achieved at different parameter settings, the fluctuation of mean average precision is typically less than 1% especially when

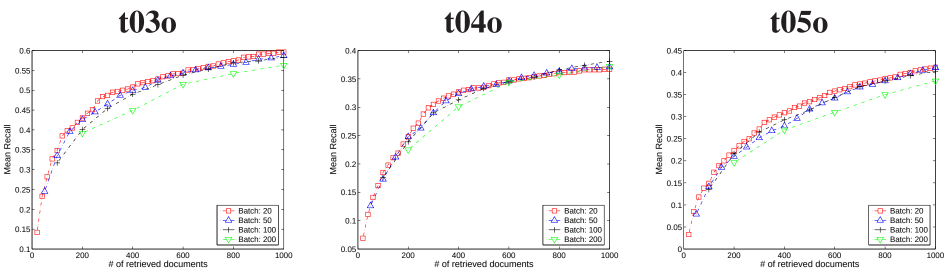


Figure 6.6: Comparison of incremental batch sizes in relevance feedback. Text retrieval outputs are used as the initial search results for pLCA.

M_D is larger than 200. It shows that pLCA is not too sensitive to the variation on the number of updated documents.

6.2.2 Results with Human Feedback

In order to evaluate the advantage of using additional human feedback, we implemented pLCA in an interactive video retrieval process called Manual Browsing with Resizing Pages (MBRP), of which the basic idea is to let human users to sequentially browse the computer-generated ranked list for each query and label as many correct documents as possible within a fixed amount of time. After every K documents being scanned, we updated the combination weight of each semantic concept based on the fixed point equations described in Eqn(6.8) using available user annotations. Then we reranked the next M_D video documents in the initial list and presented them to the user. In this paper, we only show the retrieval results with $K = 50$ and $M_D = 400$ due to the space limit. Moreover, to facilitate the labeling effort, we used the officially provided ground truth as a surrogate to simulate the human labeling process.

Figure 6.5 presents a performance comparison of pLCA with and without human feedback. Similarly, the experiments are carried out on a total of 9 configurations that combine three video collections and three initial retrieval outputs. We adopt mean recall as our performance criterion, because it is essentially the same as mean average precision when all positive video documents are assumed to be ranked at the top. From the figure, we can find that human feedback, once available, can always be helpful to boost the retrieval results of pLCA, which demonstrates the benefits of incorporating manual relevance judgment. Among all the video collections, the TREC'03 collection gain the most benefits from the feedback process, followed by the TREC'05 collection. Moreover, the performance improvement using human feedback is more significant when pLCA is applied on a lower baseline setting, e.g., text retrieval, as compared with a higher baseline setting, e.g., ApLQA. Figure 6.6 compares the mean recall of pLCA with various incremental batch sizes in the relevance feedback process. It shows that adjusting the combination parameter with a smaller batch of examples each step can be more effective than adjusting with a larger batch. However, if we re-estimate the parameters too frequently, this will also introduce a higher cost on the learning procedure. In practice, we need to find a trade-off to balance the learning cost and performance. Finally, we want to emphasize that these experiments were designed to evaluate the advantage of utilizing human feedback in pLCA and thus they are not compared with other relevance feedback algorithms. Such kinds of comparisons will be conducted in the future work.

6.3 Conclusion

In this chapter, we propose an automatic multimedia retrieval approach called probabilistic local context analysis(pLCA), which can automatically leverage useful high-level semantic concepts to refine the initial retrieval results. This approach can be represented as an undirected graphical model by treating document relevances and combination weights of semantic concepts as latent variables. Thus, it allows the information from initial retrieval results to influence the selection of semantic concepts for the given query. Built on a sound probabilistic foundation, pLCA is effective for improving video retrieval without either learning a number of training data or assuming most top-ranked documents to be relevant. As an extension, we also propose a variant of the pLCA approach that can take human relevance feedback into account. Our experiments on three TREC collections have demonstrated the advantage of the proposed pLCA approaches which can achieve noticeable gains in terms of mean average precision over various baseline methods. We expect that future investigation on designing better potential functions and introducing external semantic knowledge sources in pLCA could result in a further improvement on the retrieval performance.

6.4 Discussions

6.4.1 Related Work for pLCA

The proposed models are related to the work that applied statistical learning algorithms to automatically improve existing ranking functions or retrieval models. For example, Collins et al. [Col00] considered a discriminative reranking approach using additional features of the trees to improve upon the initial ranking for natural language parsing. The initial ranking is defined by a base parser, which produces a number of candidate parse results associated with their probabilities. Then a second model is applied to update the ranking with additional tree features. This approach allows the parsing tree to be represented by an arbitrary number of features no matter how they interact or overlap with each other. Tieu et al. [TV01] used boosting algorithms to choose a small number of features from millions of highly selective features for image retrieval. First, a user provides a few query images as examples and the AdaBoost algorithm is used to collect a very small set of critical features. Since only a small number of image features are extracted, its query process is quite efficient. Blum et al. [BM98] proposed the co-training algorithm which iteratively learns noise-tolerant models using the noisy labels provided by another classifier. By assuming the two views are independent of each other, co-training can augment the prediction from a small number of labeled data by incorporating extra information from unlabeled data. In the task of collaborative filtering, Freund et al. [FISS98] proposed the RankBoost algorithm which learns to rank a set of objects by combining multiple "weak" classifiers to build up a more accurate composite classifiers. Taskar et al. [TWK03] developed a learning method based on undirected probabilistic models to induce and utilize "unseen" features in the testing set. Their approach introduced a continuous hidden variable for each unseen feature to describe its influence on the class. The probabilistic inference over the test data can provide an estimation over the distribution of hidden variables and thus outperform the learning method using the training data alone.

The learning process of pLCA also bears resemblances to the retrieval algorithms called pseudo-relevance feedback (PRF) or local context analysis in text retrieval [CYF+97, XC00], of which the essence is to utilize top-ranked retrieved documents as evidence to select informative terms for query expansion. In this approach, a small number of top-ranked documents are assumed to be relevant and used in a relevance feedback process to modify the query description. The similar approach was also applied in the domain of multimedia retrieval [YHJ03] by learning with the most irrelevant image examples as negative data and the query images as positive data. However, in some cases, these approaches might deteriorate the retrieval performance especially when their underlying assumption is violated, i.e., most of the top-ranked documents have to be relevant to the query.

6.4.2 Relevance Feedback

The traditional relevance feedback procedure includes extracting terms from the relevant documents provided by users and appending the additional terms to the query. One of the earliest relevance feedback algorithms was proposed by J.J. Rocchio [Roc71]. The feedback iterations modify the query vectors by iteratively increasing the weights of terms contained in positive documents and penalizing the terms in negative documents. Many extensions such as Ide regular algorithm and Ide dec-hi algorithm [Ide69] are proposed based on Rocchio algorithm. The recent advancement of machine learning introduced new alternatives for the relevance feedback algorithms. Support vector machines(SVMs) [DSG01] has been applied for relevance feedback and achieved much better performance than Rocchio especially at the beginning of the feedback iterations. Instead of the explicit feedback, White et al. [WJR05] consider the form of implicit feedback which monitors searcher interaction with different representations of top-ranked documents and chooses new retrieval strategies accordingly.

Compared with text retrieval, relevance feedback has become more popular in the domain of image retrieval, partially due to the inherent difficulties of defining similarity functions on low-level image features. Early approaches usually fall in the categories of "query point movement" and "query reweighting" where for each iteration, a better query point are found together with a re-weighting of individual feature dimensions. MARS [RHM97], among one of the first image retrieval systems considering relevance feedback, updated the query point using a Rocchio-like algorithm and re-weight the distance metric with inverse variance of feedback data. MindReader [ISF98] proposed a well-founded theoretical framework for minimizing the distance between query vectors and positive feedback examples. These algorithms can interpreted from a probabilistic viewpoint, which are equivalent to optimizing the data log-likelihood given the feedback examples are drawn from Gaussian distributions. Recent relevance feedback approaches applies more advanced machine learning tools such as SVMs [CHV99] and kernel biased discriminant analysis(KBDA) [ZH01].

Chapter 7

Beyond Multimedia: Meta-Search on Text Collections

The proposed combination approaches have been shown to be effective to improve the multimedia retrieval performance, but the applicability of these approaches is not only limited to the domain of multimedia data analysis. To demonstrate the effectiveness of the proposed approaches beyond multimedia data, we further extend our retrieval experiments to a meta-search task on large-scale text collections, of which the goal is to combine the outputs from multiple search engines to form a better ranked list. In the following, we first briefly review two of the proposed approaches, i.e., probabilistic latent query analysis(pLQA) and probabilistic local context analysis(pLCA). Then we report their retrieval performance on the meta-search task.

7.1 Review: probabilistic latent query analysis

The query analysis approaches aim to adapt the combination functions for each unseen query by learning from past retrieval results. However, given the virtually infinite number of unseen queries, it is impractical to learn the combination function simply on a per query basis. To solve this, we propose an approach called *probabilistic latent query analysis*(pLQA) inspired by the algorithm *probabilistic latent semantic analysis*(pLSI) [Hof99], with the goal of automatically discovering the mixing structure of the query space without explicitly defining query classes. To achieve this, we make the following assumptions in our models: (1) the entire query space can be described by a finite number of query classes, where queries from each class share the same combination function; (2) the query description can be used to indicate which class a query belongs to. Thus, we have derive the joint probability of relevance y and latent variable z as,

$$P(y_+, z|Q, D; \mu, \lambda) \;\; = \;\; P(z|Q; \mu)P(y_+|Q, D, z; \lambda), \tag{7.1}$$

where μ is the parameter for multinomial distributions, λ is the combination parameter for query classes. This model is referred to as the *basic pLQA* (**BpLQA**) model.

However, discovering the underlying structure of query space by itself is not sufficient to handle the retrieval source combination, because a practical combination model should be able to predict combination parameters for unseen queries outside the training collection. Unfortunately, BpLQA cannot easily generalize the multinomial parameters μ to any of these unseen queries, because each parameter $\mu_{\cdot t}$ in BpLQA specifically corresponds to the t^{th} training query. To address this problem, we propose an adaptive approach aiming at parameterizing the mixing proportion $P(z|Q_t; \mu)$ using a specific set of features directly extracted from query topics, or called query features. They are able to capture important characteristics of users' information need. Formally, we can represent each query as a bag of query features $\{q_1, ...q_L\}$. The mixing proportions $P(z_k|Q; \mu)$ can then be modeled using a soft-max function $\frac{1}{Z}\exp(\sum_l \mu_{zl} q_l)$, where $Z = \sum_z \exp(\sum_l \mu_{zl} q_l)$ is the normalization factor that scales the exponential function to be a probability distribution. By substituting the mixing proportion, the BpLQA model can be generalized into the adaptive pLQA model(**ApLQA**) as,

$$P(y_+|Q, D) = \frac{1}{Z}\sum_z \exp(\sum_l \mu_{zl} q_l)\sigma\left(\sum_{i=1}^{N} \lambda_{zi} f_i(Q, D)\right). \tag{7.2}$$

These formulations offer a probabilistic interpretation for latent query types, provides guideline to estimate number of query types and allows the mixing of multiple query types in a single query. A further extension of pLQA is called *hierarchical pLQA* model (HpLQA) that can model the distributions of query-specific combination components in a single query class via a hierarchical Bayesian model.

7.2 Review: probabilistic local context analysis

Query analysis offers a useful way to incorporate query information into the knowledge source combination, e.g., learning query independent combination models and query-class based combination models. However, since these learning approaches can only capture general patterns that distinguish relevant and irrelevant training documents, their power is usually limited by the number of available manual relevance judgments. We find that it is more desirable to develop retrieval approaches that can adaptively leverage unweighted semantic concepts on a per query basis without the support of training data.

In order to automatically leverage additional semantic concepts, we propose a novel retrieval model called probabilistic local context analysis(pLCA) by considering the combination weights of unweighted semantic concepts as latent variables in an undirected graphical model. In analogy to a conditional random field [LMP01], we can derive the conditional probability of relevance $\mathbf{y}$, latent weights ν given initial weights λ and documents $\mathbf{D}$ as follows,

$$p(\mathbf{y}, \nu | \lambda, \mathbf{D}) \propto \prod_{l} p_0(\nu_l) \prod_{j=1}^{M_D} \exp\left(y_j \sum_{i \in W} \lambda_i f_i(D_j) + y_j \sum_{l \in U} \nu_l f_l(D_j) \right). \quad (7.3)$$

where $W = \{i : \lambda_i \neq 0\}$ contains the indices of initially weighted semantic concepts, $U = \{i : \lambda_i = 0\}$ contains the indices of unweighted semantic concepts, ν_l is a latent combination weight for the l^{th} unweighted concepts. The prior distribution $p_0(\nu_l)$ represents how likely it is that an unweighted semantic concept is relevant to the user's information need only based on the query description. For example, the query of "finding the map of Iraq" can induce a high weight on the semantic concept of "map". However, this prior term is not able to capture the semantic concepts that are not explicitly mentioned in the query. Therefore, to further refine the document relevances, we use the potential $\exp(y_j \lambda_i f_i(D_j))$ to capture the effects from initial combination parameters and the potential $\exp(y_j \nu_l f_l(D_j))$ to model the connections from additional unweighted concepts. Since both λ_i and $f_i(D_j)$ are already given, we can pre-compute the initial retrieval results $f^{\lambda}(D_j) = \sum_i \lambda_i f_i(D_j)$ and simplify the Eqn(7.3) to be,

$$p(\mathbf{y}, \nu | \lambda, \mathbf{D}) \propto \prod_{l} p_0(\nu_l) \prod_{j=1}^{M_D} \exp\left(y_j f^{\lambda}(D_j) + y_j \sum_{l \in U} \nu_l f_l(D_j) \right). \quad (7.4)$$

Because of the presence of the log-partition function in the undirected graphical model, it is usually intractable to compute the posterior probability with an exact inference approach. Therefore, we resort to variational methods to provide an approximate inference for the intractable posterior distributions. Specifically we can derive the following fixed point equations,

$$\gamma_j = \left[1 + \exp\left(2f_j^{\lambda} + 2\sum_l \beta_l f_{jl}\right)\right]^{-1},$$

$$\beta_l = \nu_l^0 + \sum_j (2\gamma_j - 1)\sigma_l^2 f_{jl}. \tag{7.5}$$

These equations are invoked iteratively until the change of KL-divergence is small enough. Upon convergence, we use the final $q(y_j|\gamma_j)$ as a surrogate to approximate the posterior probability $p(y_j|\mathbf{a}, \mathbf{D})$ without explicitly computing the integral. Since $q(y_j|\gamma_j)$ is a Bernoulli distribution, we can simply rank the documents in an descending order of the parameter γ_j as the retrieval outputs.

7.3 Experiments on Meta-Search

The following experiments are designed based on the task of meta-search that combines multiple search engines on a single text collection. The TREC-8 collection [VH99], which is the latest in a series of workshops designed to foster research in text retrieval, is used as our testbed which contains 50 query topics and around 2GB worth of documents. Each topic consists of both a short topic title and a long topic description. 66 groups from 16 different countries were participated in the competition. The diversity of the participating groups have made TREC represent many different methods for retrieval. There were 129 ad-hoc runs submitted for the task, including 13 manual runs and 116 automatic runs. From the submitted outputs provided by all the participants, we extracted the top five manual retrieval systems and top five automatic retrieval systems as inputs of the meta search system. Their system codes are READWARE2, orcl99man, 8manex, CL99XTopt, iit99ma1, pir9Attd, att99atde, ibms99a, ok8amxc, and fub99td respectively. Each system has at best returned 1000 documents for each query. The relevance judgment was officially provided by NIST using a pooling method. We adopt the sum normalization scheme [MA01] which normalizes the sum of scores from each submission runs to be one and shifts the minimum to be zero. We also extracted the following query features for learning the ApLQA model:

- Length of the query title;
- Appearance of named entities in the query;

Method	MAP	P30	P100	R1k
BOU	0.465(+0%)	0.587	0.414	0.645
CombSUM	0.565(+21%)	0.603	0.420	0.932
CombMNZ	0.536(+15%)	0.603	0.396	0.914
QInd	0.587(+26%)	0.609	0.441	0.914
ApLQA	0.608(+30%)	0.605	0.450	0.942
Oracle	0.660(+42%)	0.634	0.463	0.946

Table 7.1: Comparison of combination approaches in meta-search.

- The score ratio between the first ranked document and 50th ranked document for each of the ten systems.

Together with a constant term, we generated a total of 13 query features for each query. These features are designed in an attempt to capture information about query difficulties and generalities.

7.3.1 Results of pLQA

In the following discussions, we examine the performance of various meta-search algorithms. The retrieval performance is averaged over the last 25 queries in terms of MAP, precision at top 30, 100 documents and recall at top 1000 documents. All of the retrieval systems are combined to generate the final retrieval output. We evaluated the best underlying retrieval system(BOU) in addition to four meta-search strategies, including CombSUM, CombMNZ, query independent combination (QInd) and ApLQA. For those algorithms that require parameter estimation(QInd and ApLQA), we use the first 25 queries as the training data. As shown in Table 7.1, CombSUM and CombMNZ can improve upon BOU by a margin of around 10% MAP. Between them, the performance of CombMNZ is slightly worse than that of CombSUM. With aid of the training set, QInd that uses flexible wights is superior to CombSUM that fixes equal weights for every retrieval system. Finally, by introducing the query features and allowing the combination weights vary across different queries, ApLQA offers an additional 2% improvement over QInd w.r.t. MAP. This advantage mainly comes from the extra combination flexibilities provided by ApLQA.

The next experiment is designed to show how well the meta-search algorithms can perform by incorporating more retrieval systems. These systems are incorporated in descending order of their individual performances, and we expect that a successful meta-search algorithm should be able to consistently improve no matter

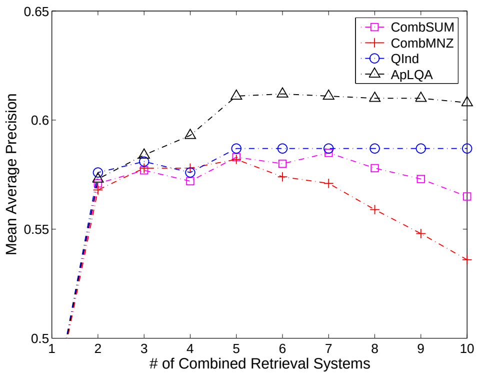


Figure 7.1: Performance of meta-search on TREC-8 vs. the number of combined systems.

how many bad systems are added. Figure 7.1 shows the comparison between different meta-search algorithms against the number of combined systems. Although their performances are on par with each other when less than three systems are combined, CombSUM and CombMNZ begin to degrade after incorporating more than four systems, because the later-added systems have worse performance but they are still assigned the same weights. In contrast, QInd and ApLQA achieve a stable performance by reducing the weights on those worse-performing systems based on the information from the training set. ApLQA benefits more from additional retrieval systems because of its flexible weight setting. This analysis gives us a clearer idea about the strengths of the proposed method as compared to CombSUM/MNZ.

7.3.2 Results of pLCA

In the following, we examine the performance of various meta-search algorithms. The retrieval performance is averaged over the last 25 queries in terms of MAP, precision at top 30, 100 documents and recall at top 1000 documents. As shown in Table 7.2, pLCA can improve the average precision around 1-2% for both sets of the initial search results, i.e., QInd and ApLQA. In this case, the less significant improvement can be partially explained by the fact that only 5 additional ranking features are available to be combined, as compared to a much larger number of features (50 ranking features) in the video collections. However, these results still confirm the effectiveness of pLCA on the meta-search task. Figure 7.2 confirms the advantage of incorporating additional human feedback into the pLCA algorithm. As expected, with more feedback documents available, the performance

Initial	pLCA	MAP	P30	P100	P1k	R30	R100	R1k
QInd	-	0.587	0.609	0.441	0.093	0.375	0.596	0.914
	Norm	0.592	0.612	0.438	0.093	0.373	0.598	0.914
ApLQA	-	0.608	0.605	0.450	0.096	0.369	0.606	0.942
	Norm	0.609	0.613	0.448	0.096	0.372	0.606	0.942

Table 7.2: Comparison of two baseline approaches and their pLCA-augmented retrieval outputs in meta-search.

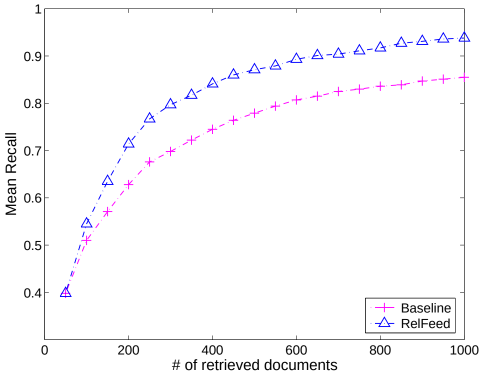


Figure 7.2: Performance of meta-search with additional human feedback. The initial results are provided by the best single retrieval system.

improvement is more considerable compared with the baseline.

7.4 Conclusions

To demonstrate the effectiveness of the proposed approaches beyond multimedia data, we further extend the retrieval experiments to a meta-search task on large-scale text collections in this section, of which the goal is to combine the outputs from multiple search engines to form a better ranked list. The experimental results have confirmed that the proposed approaches (i.e., pLQA and pLCA) can also produce better performance in another type of data collections other than multimedia data. In practice, the applicability of the proposed methods can be extended to many other areas such as question answering, web IR, cross-lingual IR, multi-sensor fusion, human tracking and so forth. We leave the explorations of these extensions into future work.

Chapter 8

Conclusions and Future Directions

In this chapter, we summarize the major research results presented in this thesis, and discuss some future directions for multimedia information retrieval.

8.1 Summary

Multimedia information retrieval systems, which aim to search a large number of multimedia data for documents relevant to an information need, offers an important platform to access and manage the vast amount of multimedia contents online. In recent years, the research community of multimedia retrieval has been gradually shifting its emphasis from analyzing one media source at a time to exploring the opportunities to select and combine diverse knowledge sources from correlated media types and context, given the rapid development of large-scale semantic concept detection techniques and retrieval approaches on various modalities. In order to develop strategies for combining multimedia knowledge sources, we need to address two major research challenges, i.e., *what* to combine (identify available knowledge sources from multimedia data) and *how* to combine (develop effective combination strategies to merge multiple knowledge sources). It has always been a significant challenge to develop principled combination approaches and capture useful factors such as query information and context information in the retrieval process.

This thesis presents a conditional probabilistic retrieval model as a principled framework to combine diverse knowledge sources. This is the first complete probabilistic model for multimedia retrieval that can handle multiple forms of ranking features, including query dependent features (uni-modal retrieval outputs) and query independent features (semantic concept indexing). It can also integrate multiple ranking features as well as query information and context information in a unified framework with a solid probabilistic foundation. In order to deal with heterogenous ranking features, a discriminative learning approach is suggested for estimating the combination parameters. The experimental results have confirmed the superiority of using discriminative models to using generative models for parameter estimation. Moreover, in order to incorporate the ranking information into the learning process, we also develop a general margin-based rank learning framework for the information retrieval task, which aims to optimize the number of discordant pairs between the predicted ranking and the target ranking. An efficient approximation is proposed for the margin-based rank learning framework which can significantly reduce the computational complexity with a negligible loss in the performance.

Under this retrieval framework, we overview and compare a number of state-of-the-art approaches for extracting ranking features from various multimedia knowledge sources. Numerous factors of text/image retrieval have been discussed in detail, including retrieval models, text sources, expansion window size, query expansion, visual features, similarity measures and their combination strategies. We also present the general approaches and discussed several open research directions for automatic semantic concept detection in multimedia collections. These studies offer a useful guideline for researchers to select the suitable algorithms to deal with difference knowledge source in multimedia systems. Meanwhile, we develop several novel machine learning approaches for extracting ranking features, e.g., *SVM ensembles* to handle rare class, *semi-supervised cross feature learning* to leverage multimodal information, *undirected graphical models* to model concept relations and *dual-wing harmoniums* to discover hidden concepts.

To incorporate the query information into the combination process, we present two type of query analysis approaches. The first type is called query-class dependent retrieval model, of which the basic idea is to first classify each query into one of the predefined classes and then apply the associated combination weights to fuse the outputs from multiple retrieval sources. The experimental results demonstrate that applying query-class dependent weights can considerably improve the retrieval performance over the query-independent weights. In order to automatically detect query classes from development data, we further propose a series of retrieval models called probabilistic latent query analysis (pLQA) to merge multiple retrieval sources, which unifies the combination weight optimization and query class categorization into a discriminative learning framework. Four pLQA models have been discussed which evolve from a basic version(BpLQA) to an adaptive version (ApLQA) that operates on the query feature space, a kernel version (KpLQA) that builds on a Mercer kernel representation, a hierarchical version that bases itself on a hierarchical Bayesian model. In contrast to the typical query-independent and query-class combination methods, pLQA can automatically discover latent query classes from the training data rather than relying on manually defined query classes. Also, it can associate one query with a mixture of query classes and thus non-identical combination weights. Finally, based on statistical model selection principles, we can estimate the number of query classes by maximizing the regularized likelihood. Our experiments in two large-scale retrieval applications, i.e., multimedia retrieval and meta-search on the TREC collections, demonstrate the superiority of the proposed methods which can achieve significant gains in average precision over the query-independent/query-class combination methods.

Although query analysis offers a useful way to incorporate the factor of query information into combination models, their power is often limited by the number of available manual relevance judgment. In order to automatically leverage useful high-level semantic concepts without training data, we propose an automatic multimedia retrieval approach called probabilistic local context analysis(pLCA). This approach can be represented as an undirected graphical model by treating document relevances and combination weights of semantic concepts as latent variables. Thus, it allows the information from initial retrieval results to influence the selection of semantic concepts for the given query. Built on a sound probabilistic foundation, pLCA is effective for improving video retrieval without either learning a number of training data or assuming most top-ranked documents to be relevant. As an extension, we also propose a variant of the pLCA approach that can take human relevance feedback into account. Our experiments on three TREC collections have demonstrated the advantage of the proposed pLCA approaches which can achieve noticeable gains in terms of mean average precision over various baseline methods.

To evaluate the performance of the proposed approaches, we provide a thorough study on the standard multimedia collections and offer baseline performances for other researchers to compare with. Our additional relevance judgement could be contributed to the information retrieval community for developing better algorithms. We also want to emphasize that although the combination approaches developed in this thesis is motivated by the multimedia retrieval problem, their contributions and potential applications are not only limited to this domain. For example, most of the proposed approaches are also examined on the task of meta-search over large-scale text collections. The applicability of the proposed methods can be extended to many other areas involving knowledge source combination, such as question answering, web IR, cross-lingual IR, multi-sensor fusion, human tracking and so forth.

8.2 Future Directions

The proposed conditional probabilistic retrieval model offers many new opportunities to develop principled retrieval approaches for combining multimedia knowledge sources especially for multimedia data. As demonstrated in previous chapters, this model allows us to incorporate heterogenous types of useful information in the retrieval process. But the approaches we describe in this thesis only reveal a small tip of the full potentials of the proposed model. Many interesting future research directions can be explored as follows,

Model knowledge relations Typically, we assume each ranking feature is independently generated from a single knowledge source. However, knowledge sources are not isolated with each other. For example, the concept "outdoors" should have connections with the concept "sky". Therefore, it is desirable to go beyond the independent assumption to model the relationship between ranking features. The tree-augmented term weighting scheme proposed by Van Rijsbergen [Rij79] can be viewed as a starting point to follow.

Handle missing features In practice, some ranking features might not always be available for entire collections, especially when the number of features is large. In this case, how to compensate the missing modalities and adjust the retrieval models accordingly will become an key issue. From another perspective, currently the semantics of ranking features have to be manually defined before the learning process, which usually requires a time-consuming human annotation process to support. It will become very interesting if we can discover latent knowledge sources within the learning process of combination models.

Refine query classes Introducing query classes into the combination model provides a practical way to handle the query information. However, the organization of query classes and the definition of query features still have room to improve. For instance, we can design a hierarchy layout for the query classes so as to capture the top-down organizations between general classes and specific classes. We can also investigate on designing better query features for ApLQA and introducing new kernels to KpLQA using pairwise distance metrics such as WordNet distance or edit distance.

Introduce user context Standard information retrieval systems treat each query in the same way without considering the user context. However, in a real retrieval scenario, a user's information need is often iteratively refined based on a series of short-term retrieval sessions and long-term preference biases. Such an iterative refinement process suggests that the retrieval outputs should depend on the previous actions that the user has taken. For example, if the user is looking for "George Bush" in the previous query and "Rice" in the current query, the latter one will have higher possibilities to be referred to "Condoleezza Rice" rather than "Rice University". Indeed, the user context can be constructed from all the information about previous actions and user environments, such as previous query keywords, explicit/implicit feedback, personal profile, query time, query location and so on. These factors can be formally incorporated into the conditional probabilistic retrieval model via a prior function. It would be very interesting to extend the proposed framework so as to optimize long-term retrieval utility over a dynamic user context.

Exploit social intelligence Recent years have seen the emergence of social intelligence in a wide range of web-based applications. The most popular form of social intelligence is Google, which ranks pages on the basis of links they get. In other words, it simply uses the brains of others. Also that is exactly what Wikipedia is doing. In doing so, it is not only competing successfully with The New York Times, but also old standard, Encyclopedia Britannica. Following the similar ideas, we could consider how to leverage the power of social intelligence to improve the information retrieval quality, such as crowd-sourcing the video concept annotation to the huge amount of web users, utilizing the external knowledge shared on the web and exploiting the social network to uncover users' inherent connections.

Apply to other domains The applicability of the proposed approaches is not limited to the multimedia retrieval problems. They can be used in many other areas involving knowledge source combination, such as question answering, web retrieval, cross-lingual information retrieval, multi-sensor fusion, human tracking and so forth. We would like to explore these opportunities in our future work.

Appendix A

Derivation of HpLQA

In this appendix, we derive the variational inference algorithm of hierarchical probabilistic latent query analysis(H-pLQA) described before. Specifically, we adopt the mean field approximation [PA87] in our derivation. The first step is to construct the following family of variational distributions,

$$q(z, w) = q(w|\gamma, V)q(z|\phi),$$

as a surrogate to approximate the posterior distribution $p(z, w|y_j, Q, D_j)$, where $q(w|\gamma, V)$ is a Gaussian distribution with mean γ and the variance V, $q(z|\phi)$ is a multinomial distribution with K parameters ϕ_k. The independence between variables in the variational distributions results in an efficient inference algorithm as shown below.

According to the Jensen's inequality, we can provide a lower bound for the log likelihood of document relevance judgements using the auxiliary variational distribution. In more detail, we can have

$$\begin{aligned}
L(\mu, \lambda) &= \sum_j \log p(y_j|Q, D_j; \mu, \lambda) \\
&= \sum_j \log \sum_z \int_w p(z|Q; \mu) p(w|z; \lambda, V) p(y_j|Q, D_j; w) dw \\
&\geq \sum_{jz} \int_w q(z, w) \log \frac{p(z|Q) p(w|z) p(y_j|Q, D_j)}{q(z, w)} dw \\
&= \sum_j E_q \left[\log p(z|Q) + \log p(w|z) + \log p(y_j|Q, D_j)\right] + M_D H(q),
\end{aligned}$$

where $E_q[f]$ refers to the expectation of $f(x)$ with respect to the distribution of $q(x)$, $H(q)$ is the entropy function. The Jensen's inequality provides a lower bound on the log-likelihood for any variational distribution $q(z, w)$, where the equality is satisfied only when $q(z, w) = p(z, w|y_j, Q, D_j)$. It could be found that the difference between the true log-likelihood and the lower bound is exactly the K-L divergence between the variational posterior distribution and true posterior distribution. Formally, we can have

$$L(\mu, \lambda) = L'(\gamma, \phi; \mu, \lambda) + \sum_j KL(q(z, w)||p(z, w|y_j, Q, D_j)).$$

The above formula have demonstrated that maximizing the $L'(\gamma, \phi; \mu, \lambda)$ w.r.t. the variational parameter γ and ϕ is the same as minimizing the KL-divergence between the variational and true posterior distribution. Therefore, we can alternatively solve a simpler optimization problem, i.e., to maximize the lower bound $L'(\gamma, \phi; \mu, \lambda)$ in order to find the best variational distributions to approximate the true posterior distribution. Thus, we substitute all the model parameters and variational parameters in $L(\mu, \lambda)$ to expand the lower bound,

$$\begin{aligned} &L'(\gamma, \phi; \mu, \lambda) \\ &= M_D \sum_k \phi_k \log \mu_k + M_D \sum_k \phi_k [-\frac{1}{2}(\lambda_k - \gamma)^T \Sigma^{-1}(\lambda_k - \gamma) - \frac{1}{2} Tr(\Sigma^{-1} V)] \\ &+ \sum_j \int_w p(w|\gamma, V) \log \sigma(y_j w^T x_j) dw - M_D \sum_k \phi_k \log \phi_k + \frac{1}{2} M_D \log |2\pi V|, \quad \text{(A.1)} \end{aligned}$$

where k is the parameter index of z, x_j is the vector of ranking features $\{f_i(Q, D_j)\}$. The constant terms in $E_q[\log p(w|z; \lambda)]$ and $E_q[\log q(w)]$ are canceled out with each other. Now the optimization difficulty lies in the third term in Eqn(A.1) (denoted as I), i.e., the integral of the logistic regression term. To address this, we appeal to the following bound using a Gaussian distribution to approximate the logistic function,

$$\sigma(a) \geq \sigma(\xi) \exp\left[\frac{a - \xi}{2} + g(\xi)(a^2 - \xi^2)\right],$$

where a is any real number, ξ is the variational parameter and $g(\xi)$ is $(1/2 -$

$\sigma(\xi))/2\xi$. By substituting this inequality into the integral I, we can get,

$$\begin{aligned} I &\geq \sum_j \int_w p(w|\gamma, V) \left[\log \sigma(\xi_j) + \frac{y_j w^T x_j - \xi_j}{2} + g(\xi_j)((w^T x_j)^2 - \xi_j^2)\right] dw \\ &= \sum_j \log \sigma(\xi_j) + \sum_j \frac{y_j \gamma^T x_j - \xi_j}{2} + \sum_j g(\xi_j)[x_j^T (V + \gamma\gamma^T) x_j - \xi_j^2]. \end{aligned}$$

Hence, we can rewrite Eqn(A.1) to be,

$$\begin{aligned} &L'(\gamma, \phi, \xi; \mu, \lambda) \\ &= M_D \sum_k \phi_k \left\{\log \mu_k - \frac{1}{2}(\lambda_k - \gamma)^T \Sigma^{-1}(\lambda_k - \gamma) - \log \phi_k - \frac{1}{2} Tr\left(\Sigma^{-1} V\right)\right\} \\ &+ \sum_j \log \sigma(\xi_j) + \sum_j \frac{y_j \gamma^T x_j - \xi_j}{2} + \sum_j g(\xi_j)[x_j^T (V + \gamma\gamma^T) x_j - \xi_j^2] + \frac{M_D}{2} \log |2\pi V|. \end{aligned}$$

In the following discussions, we show how to derive the variational update rules based on the maximization of above lower bound w.r.t. the parameters of γ, ϕ, ξ. Let us first optimize L' w.r.t. the logistic variational parameter ξ_j. Taking the derivative of L' with respect to ξ_j yields,

$$\frac{\partial L'}{\partial \xi_j} = \frac{\partial g(\xi_j)}{\partial \xi_j}[x_j^T (V + \gamma\gamma^T) x_j - \xi_j^2].$$

Setting this derivative to be zero, we have the optimal value of ξ_j as

$$\xi_j = [x_j^T (V + \gamma\gamma^T) x_j]^{1/2}.$$

Next, we go ahead to maximize the L' w.r.t. the k^{th} parameter of latent variable z, ϕ_k. Note that the parameters ϕ_k need to satisfy the constraint that $\sum_k \phi_k = 1$. By dropping the terms irrelevant to ϕ and adding the Lagrange multipliers, we can form the following Lagrangian,

$$L'_\phi = M_D \sum_k \phi_k \left\{\log \mu_k - \frac{1}{2}(\lambda_k - \gamma)^T \Sigma^{-1}(\lambda_k - \gamma) - \log \phi_k\right\} + \tau(\sum_k \phi_k - 1),$$

where τ is the Lagrangian parameter. We can get the optimal ϕ by letting the derivative to be zero,

$$\phi_k \propto \mu_k \exp[-\frac{1}{2}(\lambda_k - \gamma)^T \Sigma^{-1}(\lambda_k - \gamma)].$$

Finally, we can maximize the L' w.r.t. the parameter γ. By taking the derivative we can have,

$$\frac{\partial L'}{\partial \gamma} = M_D \sum_k \phi_k [\Sigma^{-1}(\gamma - \lambda_k)] + \frac{1}{2} \sum_j y_j x_j + 2 \sum_j g(\xi_j) x_j x_j^T \gamma.$$

Setting the equation to be zero gives a maximum at,

$$\gamma = \left[M_D \Sigma^{-1} - 2 \sum_j g(\xi_j) x_j x_j^T \right]^{-1} (\sum_k M_D \phi_k \Sigma^{-1} \lambda_k + \frac{1}{2} \sum_j y_j x_j).$$

Similarly, the variational parameters for variance is

$$V = \left(\Sigma^{-1} - \frac{2}{M_D} \sum_{i=1}^{n} g(\xi_j) x_i x_i^T \right)^{-1}.$$

These fixed point equations are invoked iteratively until the change of KL-divergence is small enough. Upon convergence, we use the resulting q as a surrogate to the original posterior p, and compute the approximate conditional probability of $p(y|Q, D)$.

In the rest of our discussions, we consider how to estimate the model parameters of μ and λ. So far, we only discuss the case for only one single query. But for the purpose of parameter estimation, we need to handle a collection of training queries $Q_1, ..., Q_{M_Q}$. Formally we want to find the model parameter μ and λ that can maximize the log-likelihood of the collection,

$$L(\mu, \lambda) = \sum_{tj} \log p(y_{tj} | Q_t, D_j; \mu_t, \lambda).$$

By applying the variational approximation again, we can achieve a similar lower bound on the log-likelihood that can be maximized w.r.t. the model parameters. The optimal parameters can be obtained by iterating the variational Expectation-Maximization(EM) process. In the E-step, the lower bound L' are maximized w.r.t. the variational parameters until the convergence as described above. In the M-step, we maximize the lower bound w.r.t. the model parameters. This process tries to find the maximum likelihood estimates under the approximate posterior provided in the E-step. The overall process correspond to a stepwise optimization in the lower bound L'.

To maximize the bound w.r.t. the latent query type parameters μ_{tk}, we can have

$$\mu_{tk} = \phi_{tk}.$$

To maximize the bound w.r.t. the combination weight parameters λ, we can obtain its derivative as,

$$\frac{\partial L'}{\partial \lambda_k} = \sum_t \phi_{tk} \Sigma^{-1} (\lambda_k - \gamma_t).$$

Setting the derivative to be zero yields,

$$\lambda_k = \left(\sum_t \phi_{tk} \gamma_t \right) / \sum_t \phi_{tk}.$$

The H-pLQA implementation can be extended to handle the query features with a modified update rule on μ,

$$\{\mu_{kl}\} \quad = \quad \arg\max_{\mu_{kl}} \sum_{t=1}^{M_Q} \sum_{k=1}^{K} \phi_{tk} \log \left[\frac{1}{Z_t} \exp(\sum_{l=1}^{L} \mu_{kl} q_{tl}) \right].$$

Appendix B

Derivation of Ranking Logistic Regression

Theorem 1. *The risk minimization estimators λ^* learned from both the margin-based rank learning framework presented in Eqn(4.13) and the ranking logistic regression algorithm presented in Eqn(4.17) are consistent with the data ranking.*

Proof: Let us first consider the Eqn(4.13). When there is a ranking feature f_a satisfies $f_a(d_j, q_t) \geq f_a(d_k, q_t), \forall q_t, \forall d_j \in D^+_{qt}, \forall d_k \in D^-_{qt}$, we can prove λ^*_a is not lower than 0 by contradiction. Assume $\lambda^*_a < 0$ in this case, since $L(\cdot)$ is monotonically decreasing, we can have

$$L\left(\sum_{i \neq a} \lambda_i f_{ijt} + \lambda^*_a f_{ajt}\right) \geq L\left(\sum_{i \neq a} \lambda_i f_{ijt} + (-\lambda^*_a) f_{ajt}\right), \forall j, t \tag{B.1}$$

where $f_{ijt} = f_i(d_j, q_t) - f_i(d_k, q_t)$ and $f_{ajt} \geq 0$ with at least one $f_{aj't'} > 0$. Therefore, this leads to a contradiction that λ^*_a is a risk minimization estimator. The case of $f_a(d_j, q) \geq f_a(d_k, q)$ can be proved similarly. This complete the proof for Eqn(4.13).

Next let us consider the Eqn(4.17). When there is a ranking feature f_a satisfies $f_a(d_j, q_t) \geq f_a(d_k, q_t), \forall q_t, \forall d_j \in D^+_{qt}, \forall d_k \in D^-_{qt}$, we are sure that the optimal $\alpha^*_i \in [\max(f_a(d_k, q_t)), \min(f_a(d_j, q_t))]$, because there are exactly $M^+_D \cdot M^-_D$ elements larger than $\min(f_a(d_j, q_t))$ and $M^+_D \cdot M^-_D$ elements smaller than $\max(f_a(d_k, q_t))$ in the union set of the right hand side of Eqn(4.16). Therefore, for all d_j, the shifted ranking feature $f^*_{ijt} = f_i(d_j, q_t) - \alpha^*_i \geq 0$. Similarly, for all d_k, the shifted ranking feature $-f^*_{ikt} \geq 0$. This recovers to the setting discussed above and thus we can have $\lambda_a^* \geq 0$. The case of $f_a(d_j, q) \geq f_a(d_k, q)$ can be proved similarly. This complete the proof for Eqn(4.17).

Theorem 2. *If* $2L(x/2) \geq L(x)$*, the inequality shown in Eqn(4.14) holds.*

Proof: We first provide a useful lemma as follows as a basis to prove the inequalities: for any $A, B \in \mathcal{R}$, based on the condition of $2L(x/2) \geq L(x)$ and the convexity of L, we can have $L(A) + L(B) \geq 2L(\frac{A+B}{2}) \geq L(A + B)$. On the other hand, we can slightly modify the lemma to be $L(A + B) + L(-A) \geq L(B)$ and $L(A + B) + L(-B) \geq L(A)$. Summing both inequalities together yields, $L(A + B) \geq \frac{1}{2}(L(A) + L(B) - L(-A) - L(-B))$. Next we go ahead to show the inequalities shown in Eqn(4.14) holds. If we set $A = f^\alpha(d_j, q) = \sum_{i=1}^n \lambda_i[f_i(d_j, q) - \alpha_i]$ and $B = -f^\alpha(d_k, q)$, both lemmas can be rewritten as,

$$L(f^\alpha(d_j, q)) + L(-f^\alpha(d_k, q)) \geq L(f(d_j, q) - f(d_k, q))$$
$$\geq \frac{1}{2}[L(f^\alpha(d_j, q)) + L(-f^\alpha(d_k, q)) - L(-f^\alpha(d_j, q)) - L(f^\alpha(d_k, q))] \quad \text{(B.2)}$$

By summing all of the cases when $\forall q_t, \forall d_j \in D_{qt}^+, \forall d_k \in D_{qt}^-$ on both sides, we can get $RR_{prox}(f) \geq RR'_{reg}(f) \geq \frac{1}{2}[RR_{prox}(f) - RR_{prox}(-f)]$.

Appendix C

Text Queries and Keywords

In the following tables, we listed the query descriptions and the corresponding text keywords used in the TRECVID'02-'05 collections. For each table, **ID** column stands for the official query identification number provided by NIST, **Auto** column stands for the automatically extracted query keywords, **Manual** column stands for the manual query keywords designed by users and **Query Description** column stands for the narrative description of query topics.

ID	Auto	Manual	Query Description
75	N/A	Eddie Rickenbacker	Eddie Rickenbacker in them
76	N/A	James H. Chandler	James H. Chandler
77	N/A	George Washington	George Washington
78	N/A	Abraham Lincoln	a depiction of Abraham Lincoln
79	N/A	leisure beach swimming sunning sand	people spending leisure time at the beach
80	N/A	musician	one or more musicians
81	N/A	football	football players
82	N/A	woman dress	one or more women standing in long dresses.
83	N/A	Golden Gate Bridge	the Golden Gate Bridge
84	N/A	Price Tower FRANK LLOYD WRIGHT	Price Tower designed by Frank Lloyd Wright and built in Bartlesville Oklahoma.
85	N/A	Washington Square Park arch New York	Washington Square Park's arch in New York City. The entire arch should be visible at some point
86	N/A	overhead views city	Overhead views of cities - downtown and suburbs. The viewpoint should be higher than the highest building visible
87	N/A	oil fields rigs derricks drilling pumping refineries	oil fields, rigs, derricks, oil drilling and pumping equipment.
88	N/A	map Unite States	a map (sketch or graphic) of the continental US.
89	N/A	butterfly	a living butterfly
90	N/A	snow moutain ridges peak	one or more snow-covered moutain peaks or ridges. Some sky must be visible.
91	N/A	parrot	one or more parrots
92	N/A	sailboats sailing ships clipper ships	one or more sailboats sailing ships clipper ships or tall ships - with some sail(s) unfurled.
93	N/A	beef cattle cows bulls herds cattle	live beef or dairy cattle individual cows or bulls herds of cattle.
94	N/A	groups people crowd urban environment streets traffic buildings	one or more groups of people a crowd walking in an urban environment (for example with streets traffic and/or buildings).
95	N/A	nuclear explosion mushroom cloud	a nuclear explosion with a mushroom cloud
96	N/A	United State flags	one or more US flags flapping
97	N/A	microscope cells microscopic cells	microscopic views of living cells
98	N/A	locomotive railroad	a locomotive (and attached railroad cars if any) approaching the viewer
99	N/A	missile rocket	a rocket or missile taking off. Simulations are acceptible

Table C.1: Query keywords for TRECVID'02

ID	Auto	Manual	Query Description
100	aerial views building road	city	aerial views containing both one or more buildings and one or more roads
101	basket basketball hoop net	NBA basketball net Michael Jordan Lakers	a basket being made - the basketball passes down through the hoop and net
102	pitcher baseball game batter	pitcher baseball batter yankee MLB	from behind the pitcher in a baseball game as he throws a ball that the batter swings at
103	Yasser Arafat	Yasser Arafat	Yasser Arafat
104	airplane	airplane aircraft plane airline airport airways continental	an airplane taking off
105	helicopter	helicopter	a helicopter in flight or on the ground
106	Tomb Cemetery Arlington National Cemetery	Tomb Cemetery Arlington National Cemetery	the Tomb of the Unknown Soldier at Arlington National Cemetery
107	rocket missile	rocket missile wardhead	a rocket or missile taking off. Simulations are acceptible
108	Mercedes Logo	Mercedes Benz Logo	the Mercedes logo (star)
109	tank	tank kuwait troops tiananmen	one or more tanks
110	diving water	diver diving	a person diving into some water
111	locomotive railroad viewer	locomotive train railroad railway Amtrack metro	a locomotive (and attached railroad cars if any) approaching the viewer
112	flame	flame fire burn	flames
113	snow mountain peaks ridge	snow mountain mountain climb mountain Everest	one or more snow-covered moutain peaks or ridges. Some sky must be visible them behind them.
114	Osama Bin Laden	Osama Bin Laden	Osama Bin Laden
115	road vehicle	road traffic block road	one or more roads with lots of vehicles
116	Sphinx	Sphinx	the Sphinx
117	people crowd urban environment	crowd riot strike panic	one or more groups of people a crowd walking in an urban environment
118	Congressman Mark Souder	Mark Souder	Congressman Mark Souder
119	Morgan Freeman	Morgan Freeman	Morgan Freeman
120	graphic dow jones industrial average rise	dow jones gain	a graphic of Dow Jones Industrial Average showing a rise for one day. The number of points risen that day must be visible.
121	mug cup coffee	coffee mocha cappuccino espresso starbucks coffeemate	a mug or cup of coffee.
122	cats	cats pets	one or more cats. At least part of both ears both eyes and the mouth must be visible.
123	John Paul Pope	John Paul Pope	Pope John Paul II
124	White House fountain	White House	the front of the White House in the daytime with the fountain running

Table C.2: Query keywords for TRECVID'03

ID	Auto	Manual	Query Description
125	street pedestrians vehicles	street pedestrians vehicles cars	a street scene with multiple pedestrians in motion and multiple vehicles in motion somewhere in the shot.
126	buildings flood water	flood storm	one or more buildings with flood waters around it/them.
127	people dogs	dogs pets	one or more people and one or more dogs walking together.
128	Congressman Henry Hyde	Henry Hyde	US Congressman Henry Hyde's face whole or part from any angle.
129	Capitol dome	capitol congress republican democrat	zooming in on the US Capitol dome.
130	hockey rink nets	hockey rink NHL	a hockey rink with at least one of the nets fully visible from some point of view.
131	fingers keys keyboard	keyboard computer laptop internet	fingers striking the keys on a keyboard which is at least partially visible.
132	people stretcher	stretcher accident hospital injured wounded	people moving a stretcher.
133	Saddam Hussein	Saddam Hussein	Saddam Hussein.
134	Boris Yeltsin	Boris Yeltsin	Boris Yeltsin.
135	Sam Donaldson	Sam Donaldson	Sam Donaldson's face - whole or part from any angle but including both eyes.
136	person golf ball hole	golf hole golfer birdie PGA tour	a person hitting a golf ball that then goes into the hole.
137	Benjamin Netanyahu	Benjamin Netanyahu	Benjamin Netanyahu.
138	people steps stairs	steps stairs stairway staircase	one or people going up or down some visible steps or stairs.
139	weapon	handheld weapon riot gun rifle shoot	a handheld weapon firing.
140	bicycles	bicycles bikes biker	one or more bicycles rolling along.
141	umbrellas	umbrella raining rainy	one or more umbrellas.
142	tennis player ball racket	tennis racket agassi sampras ATP WTA	a tennis player contacting the ball with his or her tennis racket.
143	wheelchairs	wheelchair	one or more wheelchairs. They may be motorized or not.
144	Bill Clinton flag	Clinton	Bill Clinton speaking with at least part of a US flag visible behind him.
145	horses	horses	one or more horses in motion.
146	skiers slalom course gate pole	skiers slalom skiing pole Winter Olympics	one or more skiers skiing a slalom course with at least one gate pole visible.
147	buildings fire flames smoke	fire flames smoke building fire	one or more buildings on fire with flames and smoke visible.
148	signs banners people march protest	slogan banners march protest demostration	one or more signs or banners carried by people at a march or protest

Table C.3: Query keywords for TRECVID'04

ID	Auto	Manual	Query Description
149	Condoleezza Rice	Condoleezza Rice Secretary	Condoleeza Rice
150	Iyad Allawi	Allawi	Iyad Allawi, the former prime minister of Iraq
151	Omar Karami	Karami	Omar Karami, the former prime minister of Lebanon
152	Hu Jintao	Hu Jintao	Hu Jintao, president of the People’s Republic of China
153	Tony Blair	Blair ‘Tony Blair”	Tony Blair
154	Mahmoud Abbas Abu Mazen	Abbas “Mahmoud Abbas” Mazen	Mahmoud Abbas also known as Abu Mazen prime minister of the Palestinian Authority
155	map Iraq Baghdad	Baghdad Mosul	a graphic map of Iraq location of Bagdhad marked - not a weather map
156	tennis	tennis ATP WTA Hewitt Roddick Agassi Safin Davenport Sharapova	tennis players on the court - both players visible at same time
157	shake hands	“shake hands” “shook hands”	people shaking hands
158	helicopter	helicopter	a helicopter in flight
159	George Bush	Bush	George W. Bush entering or leaving a vehicle (e.g. car van airplane helicopter etc)
160	fire flame smoke	fire flame	something (e.g. vehicle aircraft building etc) on fire with flames and smoke visible
161	banners signs	slogan banners march demostration	people with banners or signs
162	people building	“enter building” “walk out” building	one or more people entering or leaving a building
163	meeting	roundtable meeting “carbinet meeting”	a meeting with a large table and more than two people
164	ship boat	ship boat ferry smuggling vessels	a ship or boat
165	basketball	basketball NBA “Yao Ming” “Phoenix Suns”	basketball players on the court
166	palm trees	palm beach	one or more palm trees
167	airplane	airplane hijack pilot delta	an airplane taking off
168	road cars	cars road	a road with one or more cars
169	tanks and military vehicles	tank “military vehicles”	one or more tanks or other military vehicles
170	building	tall building skyscraper	a tall building (with more than 5 floors above the ground)
171	soccer goal	“winning goal” “World Cup” “European Cup” “Union Cup””	a goal being made in a soccer match
172	office table computer	office computer	an office setting i.e. one or more desks/tables and one or more computers and one or more people

Table C.4: Query keywords for TRECVID’05

Bibliography

[ACM$^+$02] J. Aslam, J. Callan, R. Manmatha, M. Sanderson, and E. Voorhees. Metasearch: Data fusion and distributed retrieval. In *Workshop on Challenges in Information Retrieval and Language Modeling*, 2002.

[AGC$^+$04] J. Adcock, A. Girgensohn, M. Cooper, T. Liu, L. Wilcox, and E. Rieffel. Fxpal experiments for trecvid 2004. In *NIST TRECVID*, 2004.

[AHI$^+$03] A. Amir, W. Hsu, G. Iyengar, C.-Y.Lin, M. Naphade, A. Natsev, C. Neti, H. J. Nock, J. R. Smith, B. L. Tseng, Y. Wu, and D. Zhang. IBM research TRECVID-2003 video retrieval system. In *NIST TRECVID-2003*, Nov 2003.

[AKJ02] S. Antani, R. Kasturi, and R. Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 4:945–65, April 2002.

[AM01] J. A. Aslam and M. Montague. Models for metasearch. In *Proc. of the 24th ACM SIGIR conference on Research and development in information retrieval*, pages 276–284, 2001.

[BBC97] M. Beigi, A. B. Benitez, and S.-F. Chang. Metaseek: A content-based meta-search engine for images. In *Proc. of SPIE*, 1997.

[BDF$^+$02] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3, 2002.

[Bea05] C. Burges and et al. Learning to rank using gradient descent. In *Proceedings of the 22nd intl. conf. on machine learning*, pages 89–96, 2005.

[Bim01] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, 2001.

[BJ03] D. Blei and M. Jordan. Modeling annotated data. In *Proc. of the 26th Intl. ACM SIGIR Conference*, 2003.

[BJF$^+$05] Steven M. Beitzel, Eric C. Jensen, Ophir Frieder, Abdur Chowdhury, and Greg Pass. Surrogate scoring for improved metasearch precision. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 583–584, New York, NY, USA, 2005. ACM Press.

[BM98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of the Workshop on Computational Learning Theory*, 1998.

[BMSW97] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *Proc. 5th Conf. on Applied Natural Language Processing*, 1997.

[BNJ03] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[BP97] A. Del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Anal. Machine Intelligence*, 19(2), 1997.

[BP98] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web*, pages 107–117, 1998.

[Bri94] E. Brill. Some advances in transformation-based part of speech tagging. In *Proc. of the 12th National Conf. Artificial Intelligence*, volume 1, 1994.

[Bur98] C.J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):955–974, 1998.

[BW99] C. Buckley and J. Walz. Smart in trec 8. In *Proc. of TREC*, 1999.

[BYRN99] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.

[CBGM97] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Region-based image querying. In *CAIVL '97: Proceedings of the 1997 Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '97)*, page 42, Washington, DC, USA, 1997. IEEE Computer Society.

[CCB95] J. P. Callan, W. B. Croft, and J. Broglio. Trec and tipster experiments with inquery. In *Proc. of the second conference on Text retrieval conference*, pages 327–343, 1995.

[CCHW05] M.-Y. Chen, M. Christel, A. Hauptmann, and H. Wactlar. Putting active learning into multimedia applications - dynamic definition and refinement of concept classifiers. In *Proceedings of ACM Multimedia*, November 2005.

[CCS00] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proc. 17th International Conference on Machine Learning(ICML00)*, pages 111–118, 2000.

[CFG$^+$04] E. Cooke, P. Ferguson, G. Gaughan, C. Gurrin, G.J.F. Jones, H. Le Borgue, H. Lee, S. Marlow, K. McDonald, M. McHugh, N. Murphy, N.E. O'Connor, N. O'Hare, S. Rothwell, A.F. Smeaton, and P. Wilkins. Trecvid 2004 experiments in dublin city university. In *NIST TRECVID*, 2004.

[CH88] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Document retrieval systems*, pages 161–171, 1988.

[CH05] M. Christel and A. G. Hauptmann. The use and utility of high-level semantic features. In *Intl. Conf. on Image and Video Retrieval (CIVR'05)*, Singapore, 2005.

[CHK+05] S.-F. Chang, W. Hsu, L. Kennedy, L. Xie, A. Yanagawa, E. Zavesky, and D.-Q. Zhang. Columbia university trecvid-2005 video search and high-level feature extraction. In *NIST TRECVID*, 2005.

[CHV99] O. Chapelle, P. Haffner, and V. Vapnik. SVMs for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1065, 1999.

[CHWC99] M. G. Christel, A. G. Hauptmann, A. Warmack, and S. A. Crosby. Adjustable filmstrips and skims as abstractions for a digital video library. In *Advances in Digital Libraries*, pages 98–104, 1999.

[CK96] Gene C.-H. Chuang and C.-C. Jay Kuo. Wavelet descriptor of planar curves: Theory and applications. *IEEE Transactions on Image Processing*, 5(1):56–70, 1996.

[CM98] M. Christel and D. Martin. Information visualization within a digital video library. *Journal of Intelligent Information Systems*, 11(3):235–257, 1998.

[CMC05] S. F. Chang, R. Manmatha, and T. S. Chua. Combining text and audio-visual features in video indexing. In *IEEE ICASSP 2005*, 2005.

[CNG+05] T.-S. Chua, S.-Y. Neo, H.-K. Goh, M. Zhao, Y. Xiao, and G. Wang. Trecvid 2005 by nus pris. In *NIST TRECVID-2005*, Nov 2005.

[CNL+04] T. S. Chua, S. Y. Neo, K. Li, G. H. Wang, R. Shi, M. Zhao, H. Xu abd S. Gao, and T. L. Nwe. Trecvid 2004 search and feature extraction task by nus pris. In *NIST TRECVID*, 2004.

[CO05] Datong Chen and Jean-Marc Odobez. Video text recognition using sequential monte carlo and error voting methods. *Pattern Recogn. Lett.*, 26(9):1386–1403, 2005.

[COH00] M.G. Christel, A.M. Olligschlaeger, and C. Huang. Interactive maps for a digital video library. *IEEE MultiMedia*, 7(1), 2000.

[Col00] M. Collins. Discriminative reranking for natural language parsing. In *Proc. of the 17th Intl. Conf. on Machine Learning*, pages 175–182, 2000.

[CR97] P. Clarkson and R. Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Proc. Eurospeech'97*, 1997.

[CRZ96] I. J. Cox, S. B. Rao, , and Y. Zhong. Ratio regions: A technique for image segmentation. In *Proceedings International Conference on Pattern Recognition*, pages 557–564, 1996.

[CS98] P. Chan and S. Stolfo. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proc. Fourth Intl. Conf. Knowledge Discovery and Data Mining*, pages 164–168, 1998.

[CS99] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proc. of EMNLP*, 1999.

[CS00] Anne Le Calv; and Jacques Savoy. Database merging strategy based on logistic regression. *Inf. Process. Manage.*, 36(3):341–359, 2000.

[CS01] K. Crammer and Y. Singer. Pranking with ranking. In *Proc. of the Advanced Neural Information Processing Systems (NIPS)*, 2001.

[CSS98] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems*, pages 451–457, Cambridge, MA, USA, 1998. MIT Press.

[CTO97] Tat-Seng Chua, Kian-Lee Tan, and Beng-Chin Ooi. Fast signature-based color-spatial image retrieval. In *ICMCS '97: Proceedings of the 1997 International Conference on Multimedia Computing and Systems (ICMCS '97)*, page 362, Washington, DC, USA, 1997. IEEE Computer Society.

[CYF$^+$97] J. G. Carbonell, Y. Yang, R. E. Frederking, R. D. Brown, Y. Geng, and D. Lee. Translingual information retrieval: A comparative evaluation. In *Proc. of IJCAI*, pages 708–715, 1997.

[DDF$^+$90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 1990.

[DKNS01] Cynthia Dwork, S. Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *World Wide Web*, pages 613–622, 2001.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[DSG01] H. Drucker, B. Shahrary, and D. C. Gibbon. Relevance feedback using support vector machines. In *Proc. 18th International Conf. on Machine Learning*, pages 122–129, 2001.

[Fag98] Ronald Fagin. Fuzzy queries in multimedia database systems. In *Proceedings of the 17th ACM symposium on Principles of database systems*, pages 1–10, 1998.

[FBF$^+$94] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic, and William Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[Fel98] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[FGJ$^+$04] C. Foley, C. Gurrin, G. Jones, H. Lee, S. McGivney, N. E. O'Connor, S. Sav, A. F. Smeaton, and P. Wilkins. Trecvid 2005 experiments in dublin city university. In *NIST TRECVID*, 2004.

[FISS98] Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. In *Proc. of the 15th Intl. Conf. on Machine Learning*, pages 170–178, San Francisco, CA, USA, 1998.

[FKS03] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 301–312, New York, NY, USA, 2003. ACM Press.

[FLN01] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. In *ACM Symposium on Principles of Database Systems*, 2001.

[Fuh92] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[Gey94] Fredric C. Gey. Inferring probability of relevance using the method of logistic regression. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–231, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[GL02] Andrew Graves and Mounia Lalmas. Video retrieval using an mpeg-7 based inference network. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346, 2002.

[GLA02] J.L. Gauvain, L. Lamel, and G. Adda. The limsi broadcast news transcription system. *Speech Communication*, 37(1-2):89–108, 2002.

[GLS95] D. Grinberg, J. Lafferty, and D. Sleator. A robust parsing algorithm for link grammars. In *Proc. of the 4th Int'l Workshop on Parsing Technologies*, 1995.

[GQXN05] J. Gao, H. Qi, X. Xia, and J.-Y. Nie. Linear discriminant model for information retrieval. In *Proceedings of the 28th international ACM SIGIR conference*, pages 290–297, New York, NY, USA, 2005. ACM Press.

[GSG$^+$03] Georgina Gaughan, Alan F. Smeaton, Cathal Gurrin, Hyowon Lee, and Kieran McDonald. Design, implementation and testing of an interactive video retrieval system. In *Proc. of 11th ACM MM Workshop on MIR*, Nov 2003.

[HAH$^+$93] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, and R. Rosenfeld. The SPHINX-II speech recognition system: an overview. *Computer Speech and Language*, 7(2):137–148, 1993.

[Hau06] A. G. Hauptmann. Automatic spoken document retrieval (1st. draft). In K. Brown, editor, *Encyclopedia of Language and Linguistics (2nd edition)*. Elsevier, 2006.

[HBC$^+$03] A. G. Hauptmann, R.V. Baron, M.-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W.-H. Lin, T. Ng, N. Moraveji, N. Papernick, C.G.M. Snoek, G. Tzanetakis, J. Yang, R. Yan, , and H.D. Wactlar. Informedia at TRECVID 2003: Analyzing and searching broadcast news video. In *Proc. of TRECVID*, 2003.

[HBHV04] Samira Hammiche, Salima Benbernou, Mohand-Sa&#239;d Hacid, and Athena Vakali. Semantic retrieval of multimedia data. In *Proc. of the 2nd ACM international workshop on Multimedia databases*, pages 36–44, 2004.

[HC04] Alexander G. Hauptmann and Michael G. Christel. Successful approaches in the trec video retrieval evaluations. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 668–675, New York, NY, USA, 2004. ACM Press.

[HCC$^+$04] A. Hauptmann, M.-Y. Chen, M. Christel, C. Huang, W.-H. Lin, T. Ng, N. Papernick, A. Velivelli, J. Yang, R. Yan, H. Yang, and H. D. Wactlar. Confounded expectations: Informedia at trecvid 2004. In *Proc. of TRECVID*, 2004.

[HCWZ01] Xian-Sheng Hua, Xiang-Rong Chen, Liu Wenyin, and Hong-Jiang Zhang. Automatic location of text in video frames. In *MULTIMEDIA '01: Proceedings of the 2001 ACM workshops on Multimedia*, pages 24–27, New York, NY, USA, 2001. ACM Press.

[HGO00] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In A. J. Smola, P. L. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 2000.

[HJN03] A. G. Hauptmann, R. Jin, and T. D. Ng. Video retrieval using speech and image information. In *Storage and Retrieval for Multimedia Databases 2003, EI'03 Electronic Imaging*, Santa Clara, CA, 2003.

[HKM$^+$97] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 762–768, 1997.

[HL98] Alexander G. Hauptmann and Danny Lee. Topic labeling of broadcast news stories in the informedia digital video library. In *Proceedings of the third ACM conference on Digital libraries*, pages 287–288. ACM Press, 1998.

[HLZ$^+$04] Jingrui He, Mingjing Li, Hong-Jiang Zhang, Hanghang Tong, and Changshui Zhang. Manifold-ranking based image retrieval. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 9–16, New York, NY, USA, 2004. ACM Press.

[HMK$^+$02] Xiaofei He, Wei-Ying Ma, Oliver King, Mingjing Li, and Hongjiang Zhang. Learning and inferring a semantic space from user's relevance feedback for image retrieval. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 343–346, New York, NY, USA, 2002. ACM Press.

[HN99] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *Journal Global Optimization*, 14, 1999.

[HO04] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *The Eleventh Symposium on String Processing and Information Retrieval (SPIRE)*, 2004.

[Hof99] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proc. of the 22nd Intl. ACM SIGIR conference*, pages 50–57, 1999.

[Hou99] R. Houghton. Named faces: putting names to faces. *IEEE Intelligent Systems*, 14(5), 1999.

[HP99] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proc. of the 16th International Joint Conference on Artificial Intelligence*, pages 688–693, 1999.

[HP02] Alexander G. Hauptmann and Norman Papernick. Video-cuebik: adapting image search to video shots. In *JCDL*, pages 156–157, 2002.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning. Springer Series in Statistics*. Springer Verlag, Basel, 2001.

[Huu05] B. Huurnink. AutoSeek: Towards a fully automated video search system. Master's thesis, University of Amsterdam, October 2005.

[HW96] A. Hauptmann and M. Witbrock. Informedia news on demand: Multimedia information acquisition and retrieval. In *Intelligent Multimedia Information Retrieval*. AAAI Press/MIT Press, Menlo Park, CA, 1996.

[Ide69] E. Ide. Relevance feedback in an automatic retrieval system. In *Report ISR-15 to the National Science Foundation*, 1969.

[IDF$^+$05] G. Iyengar, P. Duygulu, S. Feng, P. Ircing, S. P. Khudanpur, D. Klakow, M. R. Krause, R. Manmatha, H. J. Nock, D. Petkova, B. Pytlik, and P. Virga. Joint visual-text modeling for automatic retrieval of multimedia documents. In *Proceedings of ACM Multimedia*, November 2005.

[ISF98] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *Proc. of the 24rd International Conference on Very Large Data Bases*, pages 218–227, 1998.

[JAK02] M.V. Joshi, R.C. Agarwal, and V. Kumar. Predicting rare classes: Can boosting make any weak learner strong? In *the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada*, July 2002.

[Jap00] N. Japkowicz. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets. Tech Rep. WS-00-05, Menlo Park, CA: AAAI Press*, 2000.

[JCL04] R. Jin, J. Y. Chai, and S. Luo. Automatic image annotation via coherent language model and active learning. In *Proceedings of ACM Multimedia*, November 2004.

[JH01] R. Jin and A.G. Hauptmann. Headline generation using a training corpus. In *Second International Conference on Intelligent Text Processing and Computational Linguistics (CICLING01)*, pages 208–215, Mexico City, Mexico, 2001.

[JH02] Rong Jin and Alexander G. Hauptmann. Using a probabilistic source model for comparing images. In *IEEE Intl. Conf. on Image Processing(ICIP)*, 2002.

[JJ94] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[JLM03] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126, 2003.

[Joa95] T. Joachims. Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning, B. Scholkopf and C. Burges and A. Smola (ed.), Springer*, 1995.

[Joa98] T. Joachims. Making large-scale support vector machine learning practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[Joa02] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD intl. conf. on knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM Press.

[KJNN00] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *Speech and Audio Processing, IEEE Transactions on*, 8(6):695–707, 2000.

[KK03] I.-H. Kang and G. Kim. Query type classification for web document retrieval. In *Proc. of the 26th ACM SIGIR*, pages 64–71. ACM Press, 2003.

[Kle98] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 668–677, 1998.

[KNC05] L. Kennedy, P. Natsev, and S.-F. Chang. Automatic discovery of query class dependent models for multimodal search. In *ACM Multimedia*, Singapore, November 2005.

[Kra04] W. Kraaij. *Variations on Language Modeling for Information Retrieval*. PhD thesis, University of Twente, 2004.

[KW71] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

[Lee96] Tai-Sing Lee. Image representation using 2d gabor wavelets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):959–971, 1996.

[Lew02] Michael Lew, editor. *Intl. Conf. on Image and Video Retrieval*. The Brunei Gallery, SOAS, Russell Square, London, UK, 2002.

[LH02] W.-H. Lin and A. G. Hauptmann. News video classification using svm-based multimodal classifiers and combination strategies. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 323–326, New York, NY, USA, 2002. ACM Press.

[LH04] W. H. Lin and A. G. Hauptmann. Merging rank lists from multiple sources in video classification. In *IEEE International Conference on Multimedia and Expo (ICME)*, Taipei, Taiwan, 2004.

[Lie03] Rainer Lienhart. Video ocr: A survey and practitioners guide. In *Video Mining*. Kluwer Academic Publisher, 2003.

[LM94] Bingcheng Li and Songde Ma. On the relation between region and contour representation. In *Proc. IEEE Int. Conf. on Pattern Recognition*, pages 352–355, 1994.

[LMP01] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th Intl. Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[LOT94] H. Lu, B. Ooi, and K. Tan. Efficient image retrieval by color contents. In *Proc. of the 1994 Int. Conf. on Applications of Databases*, 1994.

[LR87] A. M. Leroy and P. J. Rousseeuw. *Robust regression and outlier detection*. Wiley Series in Probability and Mathematical Statistics, New York: Wiley, 1987, 1987.

[LR02] Xin Li and Dan Roth. Learning question classifiers. In *COLING'02*, Aug 2002.

[LTC97] Z. Lei, T. Tasdizen, and D.B. Cooper. Object signature curve and invariant shape patches for geometric indexing into pictorial databases. In *Proc. of Multimedia Storage and Archiving Systems II*, Dallas, TX, 1997.

[LTS03] C. Lin, B. Tseng, and J. Smith. VideoAnnEx: IBM MPEG-7 annotation tool for multimedia indexing and concept learning. In *IEEE International Conference on Multimedia and Expo*, 2003.

[LVF03] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using cotraining. In *Proc. of the Intl. Conf. on Computer Vision*, 2003.

[LWW00] Jia Li, James Z. Wang, and Gio Wiederhold. Irm: integrated region matching for image retrieval. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 147–156, New York, NY, USA, 2000. ACM Press.

[LZ03] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In *Language Modeling for Information Retrieval, Kluwer International Series on Information Retrieval*, volume 13. Springer, 2003.

[MA85] Amar Mitiche and Jake K. Aggarwal. Image segmentation by conventional and information-integrating techniques: a synopsis. *Image and Vision Computing*, 3(2):50–62, 1985.

[MA01] M. Montague and J. A. Aslam. Relevance score normalization for metasearch. In *Proceedings of the 10th international ACM CIKM conference*, pages 427–433, New York, NY, USA, 2001.

[Man02] R. Manmatha. Multimedia indexing and retrieval. In *Workshop on Challenges in Information Retrieval and Language Modeling*, 2002.

[MFR01] R. Manmatha, F. Feng, and T. Rath. Using models of score distributions in information retrieval. In *Proc. of the 27th ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

[MH79] D. Marr and E. Hildreth. Theory of edge detection. In *Proc. of Royal Society of London Bulletin*, pages 301–328, 1979.

[MKL97] Babu M. Mehtre, Mohan S. Kankanhalli, and Wing Foon Lee. Shape measures for content based image retrieval: a comparison. *Inf. Process. Manage.*, 33(3):319–337, 1997.

[MM95] W. Y. Ma and B. S. Manjunath. A comparison of wavelet transform features for texture image annotation. In *ICIP '95: Proceedings of the 1995 International Conference on Image Processing (Vol.2)-Volume 2*, page 2256, Washington, DC, USA, 1995. IEEE Computer Society.

[MM96] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.

[MMK02] I. Muslea, S. Minton, and C. A. Knoblock. Active semi-supervised learning = robust multi-view learning. In *Proc. of Intl. Conf. on Machine Learning*, 2002.

[MMM97] A. Merlino, D. Morey, and M. Maybury. Broadcast news navigation using story segmentation. In *Proc. of ACM Multimedia*, 1997.

[MS05] K. McDonald and A.F. Smeaton. A comparison of score, rank and probability-based fusion methods for video shot retrieval. In *International Conference on Image and Video Retrieval(CIVR)*, 2005.

[Nal04] R. Nallapati. Discriminative models for information retrieval. In *Proc. of the 27th SIGIR conference on Research and development in information retrieval*, pages 64–71, 2004.

[Nev86] R. Nevatia. Image segmentation. In K.S. Fu T.Y. Young, editor, *Handbook of Pattern Recognition and Image Processing*. Academic Press, San Diego, CA, 1986.

[NG00] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proc. of CIKM*, pages 86–93, 2000.

[NJ02] A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*, 2002.

[NKFH98] M. R. Naphade, T. Kristjansson, B. Frey, and T.S. Huang. Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems. In *Proc. of ICIP*, 1998.

[NKH00] M. R. Naphade, I. Kozintsev, and T. S. Huang. On probabilistic semantic video indexing. In *Proceedings of Neural Information Processing Systems*, volume 13, pages 967–973, Denver, CO, Nov. 2000.

[NPZ01] Chong-Wah Ngo, Ting-Chuen Pong, and Hong-Jiang Zhang. On clustering and retrieval of video shots. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 51–60, New York, NY, USA, 2001. ACM Press.

[NS03] A. Natsev and J. R. Smith. Active selection for multi-example querying by content. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2003.

[NS04a] M. R. Naphade and J. R. Smith. Active learning for simultaneous annotation of multiple binary semantic concepts. In *Proceedings of IEEE International Conference On Multimedia and Expo (ICME)*, pages 77–80, 2004.

[NS04b] M. R. Naphade and J. R. Smith. On the detection of semantic concepts at trecvid. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 660–667, New York, NY, USA, 2004. ACM Press.

[NST$^+$06] Milind Naphade, John R. Smith, Jelena Tesic, Shih-Fu Chang, Winston Hsu, Lyndon Kennedy, Alexander Hauptmann, and Jon Curtis. Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006.

[NT92] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. In *Proceedings of the IFIP TC2/WG 2.6 Second Working Conference on Visual Database Systems II*, pages 113–127. North-Holland, 1992.

[NZKC06] Shi-Yong Neo, Jin Zhao, Min-Yen Kan, and Tat-Seng Chua. Video retrieval using high level features: Exploiting query matching and confidence-based weighting. In *Proceedings of the Conference on Image and Video Retrieval (CIVR)*, 2006.

[OC03] P. Ogilvie and J. Callan. Combining document representations for known-item search. In *Proceedings of the 26thACM SIGIR conference on Research and development in informaion retrieval*, pages 143–150, 2003.

[PA87] C. Peterson and J.R. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[PC98] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. of the 21st ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, 1998.

[PC01] D. Pierce and C. Cardie. Limitations of co-training for natural language learning from large datasets. In *Proc. of EMNLP*, 2001.

[PF98] P. Perona and W. Freeman. A factorization approach to grouping. *Lecture Notes in Computer Science*, 1406:655, 1998.

[PGKK05] B. Pytlik, A. Ghoshal, D. Karakos, and S. Khudanpur. Trecvid 2005 experiment at johns hopkins university: Using hidden markov models for video retrieval. In *NIST TRECVID*, 2005.

[PHB97] J. Puzicha, T. Hofmann, and J.M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 267 – 272, 1997.

[Pla99] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In B. Scholkopf A. Smola, P. Bartlett and D. Schuurmans, editors, *Advances in Large Margin Classiers*. MIT Press, 1999.

[PP93] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 1993.

[PPO92] R. C. Dubes P. P. Ohanian. Performance evaluation for four classes of texture features. *Pattern Recognition*, 25(2):819–833, 1992.

[Pro00] Foster Provost. Machine learning from imbalanced data sets 101/1. In *AAAI Workshop on Learning from Imbalanced Data Sets. Tech Rep. WS-00-05, Menlo Park, CA: AAAI Press*, 2000.

[PYFD04] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Gcap: Graph-based automatic image captioning. In *Proc. of the 4th International Workshop on Multimedia Data and Document Engineering (MDDE 04), in conjunction with Computer Vision Pattern Recognition Conference (CVPR 04)*, 2004.

[PZ99] Greg Pass and Ramin Zabih. Comparing images using joint histograms. *Multimedia System*, 7(3):234–240, 1999.

[QF93] Y. Qiu and H.-P. Frei. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference*, pages 160–169, New York, NY, USA, 1993. ACM Press.

[Rea04] M. Rautiainen and et al. TRECVID 2004 experiments at mediateam oulu. In *Proc. of TRECVID*, 2004.

[RHC97] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image retrieval: Past, present, and future. In *International Symposium on Multimedia Information Processing*, 1997.

[RHM97] Y. Rui, T. Huang, and S. Mehrotra. Content-Based image retrieval with relevance feedback in MARS. In *Proc. IEEE Intl. Conf. on Image Processing*, pages 815–818, 1997.

[Rij79] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 1979.

[RJ77] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Informaiton Science*, 27, 1977.

[RJ04] Lawrence A. Rowe and Ramesh Jain. Acm sigmm retreat report on future directions in multimedia research. In *Proceedings of ACM Multimedia*, March 2004.

[RM95] L. Ramshaw and M. Marcus. Text chunking using transformation-based learning. In *Proc. of the ACL Third Workshop on Very Large Corpora*, 1995.

[Rob77] S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.

[Roc71] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[ROS04] M. Rautiainen, T. Ojala, and T. Seppanen. Cluster-temporal browsing of large news video databases. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2004.

[RS96] R. Rickman and J. Stonham. Content-based image retrieval using color tuple histograms. In *Storage and Retrieval for Image and Video Databases (SPIE)*, 1996.

[RS03] M. Elena Renda and Umberto Straccia. Web metasearch: rank vs. score based rank aggregation methods. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 841–846, New York, NY, USA, 2003. ACM Press.

[RSH96] Y. Rui, A. She, and T. Huang. Modified fourier descriptors for shape representation – a practical approach. In *Proc. of First International Workshop on Image Databases and Multimedia Search.*, 1996.

[RW94] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. of the 17th ACM SIGIR*, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[RWHB^{+}92] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC4. In *Text REtrieval Conference*, pages 21–30, 1992.

[Sal89] G. Salton. *Automatic text processing*. Addison-Wesley, 1989.

[SB91] M. J. Swain and D. H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, 1991.

[SB96] Sudeep Sarkar and Kim L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. In *IEEE Computer Vision and Pattern Recognition(CVPR)*, pages 478–483, 1996.

[SBM96] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proc. of the 19th ACM SIGIR*, pages 21–29, New York, NY, USA, 1996. ACM Press.

[SBM05] M. Srikanth, M. Bowden, and D. Moldovan. Lcc at trecvid 2005. In *NIST TRECVID*, 2005.

[SC96a] John R. Smith and Shih-Fu Chang. Automated binary texture feature sets for image retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2239–2242, 1996.

[SC96b] John R. Smith and Shih-Fu Chang. Tools and techniques for color image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 426–437, 1996.

[SC96c] John R. Smith and Shih-Fu Chang. Visualseek: A fully automated content-based image query system. In *ACM Multimedia*, pages 87–98, 1996.

[SC02] L. Si and J. Callan. Using sampled data and regression to merge search engine results. In *Proc. of the 25th ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, 2002.

[Sch78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[SF94] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *Text REtrieval Conference*, 1994.

[SKHS98] Toshio Sato, Takeo Kanade, Ellen Hughes, and Michael Smith. Video ocr for digital news archives. In *IEEE Workshop on Content-Based Access of Image and Video Databases(CAIVD'98)*, pages 52 – 60, January 1998.

[SLN^{+}02] J. R. Smith, C. Y. Lin, M. R. Naphade, P. Natsev, and B. Tseng. Advanced methods for multimedia signal processing. In *Intl. Workshop for Digital Communications IWDC*, Capri, Italy, 2002.

[SLZ01] Zhong Su, Stan Li, and Hongjiang Zhang. Extraction of feature subspaces for content-based retrieval using relevance feedback. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 98–106, New York, NY, USA, 2001. ACM Press.

[SM97] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[SM98] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998.

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[SM01] J. Sese and S. Morishita. Rank aggregation method for biological databases. *Genome Informatics*, 12, 2001.

[Smi03] J. R. Smith. Video indexing and retrieval using MPEG-7. In B. Furht and O. Marques, editors, *The Handbook of Image and Video Databases: Design and Applications*. CRC Press, 2003.

[Smo86] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, 1986.

[SO95] Markus A. Stricker and Markus Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

[SO03] A.F. Smeaton and P. Over. TRECVID: Benchmarking the effectiveness of information retrieval tasks on digital video. In *Proc. of the Intl. Conf. on Image and Video Retrieval*, 2003.

[SP02] M. Szummer and R. Picard. Indoor-outdoor image classification. In *IEEE International Workshop in Content-Based Access to Image and Video Databases, Bombay, India*, Jan 2002.

[SSL02] N. Serrano, A. Savakis, and J. Luo. A computationally efficient approach to indoor/outdoor scene classification. In *International Conference on Pattern Recognition, Qubec City, Canada*, Aug. 2002.

[Str94] Markus A. Stricker. Bounds for the discrimination power of color indexing techniques. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 15–24, 1994.

[SWG$^+$04] C.G.M. Snoek, M. Worring, J.M. Geusebroek, D.C. Koelma, and F.J. Seinstra. The mediamill trecvid 2004 semantic viedo search engine. In *Proc. of TRECVID*, 2004.

[SWS$^+$00] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval: the end of the early years. *IEEE transactions Pattern Analysis Machine Intelligence*, 22 - 12:1349 – 1380, 2000.

[SWS05] C.G.M. Snoek, M. Worring, and A.W.M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of ACM Multimedia*, November 2005.

[SWvG$^+$06] Cees G.M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W.M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of ACM Multimedia*, 2006.

[SZ03] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1470, Washington, DC, USA, 2003. IEEE Computer Society.

[TC01] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM Intl. Conf. on Multimedia*, pages 107–118, 2001.

[TFC03] Ch. Tsinaraki, E. Fatourou, and S. Christodoulakis. An ontology-driven framework for the management of semantic metadata describing audiovisual information. In *Proc. of the 15th Intl. Conf. on Advanced Information Systems Engineering (CAiSE)*, 2003.

[TG99] Tinne Tuytelaars and Luc J. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *VISUAL '99: Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 493–500, London, UK, 1999. Springer-Verlag.

[TNP96] K. Thyagarajan, J. Nguyen, and C. Persons. A maximum likelihood approach to texture classification using wavelet transform. In *International Conference on Image Processing (ICIP)*, pages 640–644, 1996.

[Tur91] H. R. Turtle. *Inference Networks for Document Retrieval*. PhD thesis, University of Massachusetts, 1991.

[TV01] K. Tieu and P. Viola. Boosting image retrieval. In *Intl. Conf. on Computer Vision*, pages 228–235, 2001.

[TWK03] B. Taskar, M. F. Wong, and D. Koller. Learning on the test data: Leveraging unseen features. In *Twentieth International Conference on Machine Learning (ICML03)*, 2003.

[Vap95] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[VC99] C. C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.

[VFJZ99] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. A bayesian framework for semantic classification of outdoor vacation images. In *SPIE Conference on Electronic Imaging, San Jose, California*, 1999.

[vG03] J.C. van Gemert. Retrieving images as text, 2003. Master Thesis, University of Amsterdam.

[VGJL95] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning collection fusion strategies. In *Proc. of the 18th ACM SIGIR conference on Research and development in information retrieval*, pages 172–179, 1995.

[VH99] Ellen M. Voorhees and Donna Harman. Overview of the eighth text retrieval conference (trec-8). In *TREC*, 1999.

[VJZ98] A. Vailaya, A. Jain, and H.J. Zhang. On image classification: City vs. landscape. In *IEEE Workshop on Content-Based Access of Image and Video Libraries, Santa Barbara, CA*, Jun 1998.

[VN06] T. Volkmer and A. Natsev. Exploring automatic query refinement for text-based video retrieval. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2006.

[WCCS04] Yi Wu, Edward Y. Chang, Kevin Chen-Chuan Chang, and John R. Smith. Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 572–579, 2004.

[WCGH99] H. Wactlar, M. Christel, Y. Gong, and A. G. Hauptmann. Lessons learned from the creation and deployment of a terabyte digital video library. *IEEE Computer*, 32(2):66–73, 1999.

[WdV04] T. Westerveld and A. de Vries. Multimedia retrieval using multiple examples. In *International Conference on Image and Video Retrieval(CIVR)*, 2004.

[Wes04] Thijs Westerveld. *Using generative probabilistic models for multimedia retrieval*. PhD thesis, CWI, Centre for Mathematics and Computer Science, 2004.

[WIB$^+$03] T. Westerveld, T. Ianeva, L. Boldareva, A. P. de Vries, and D. Hiemstra. Combining infomation sources for video retrieval: The lowlands team at TRECVID 2003. In *NIST TRECVID-2003*, Nov 2003.

[WJR05] R. W. White, J. M. Jose, and I. Ruthven. An implicit feedback approach for interactive information retrieval. *Information Processing and Management*, 2005.

[WP01] G.M. Weiss and F. Provost. The effect of class distribution on classifier learning. Technical report, Department of Computer Science, Rutgers University, 2001.

[WRZH04] M. Welling, M. Rosen-Zvi, and G. Hinton. Exponenetial family harmoniums with an application to information retrieval. In *Advance Neural Information Processing Systems*, volume 17, 2004.

[WTS04] Y. Wu, B. L. Tseng, and J. R. Smith. Ontology-based multi-classification learning for video concept detection. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2004.

[XC00] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transaction Information System*, 18(1):79–112, 2000.

[XKC$^+$04] L. Xie, L. Kennedy, S.F. Chang, A. Divakaran, H. Sun, and C. Y. Lin. Discovering meaningful multimedia patterns with audio-visual concepts and associated text. In *IEEE International Conference on Image Processing (ICIP)*, Singapore, Oct 2004.

[XYH05] E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images using dual-wing harmoniums. In *Uncertainty in Artifical Intelligence (UAI)'05*, 2005.

[YCH04] J. Yang, M. Y. Chen, and A. G. Hauptmann. Finding person x: Correlating names with visual appearances. In *Intl. Conf. on Image and Video Retrieval (CIVR'04)*, Ireland, 2004.

[YCZ+03] Hui Yang, Lekha Chaisorn, Yunlong Zhao, Shi-Yong Neo, and Tat-Seng Chua. VideoQA: question answering on news video. In *Proc. of the 11th ACM MM*, pages 632–641, 2003.

[YH03] R. Yan and A. G. Hauptmann. The combination limit in multimedia retrieval. In *Proc. of the eleventh ACM international conference on Multimedia*, pages 339–342, 2003.

[YH04] R. Yan and A. G. Hauptmann. Multi-class active learning for video semantic feature extraction. In *Proceedings of IEEE International Conference On Multimedia and Expo (ICME)*, pages 69–72, 2004.

[YH06a] R. Yan and A. G. Hauptmann. Efficient margin-based rank learning algorithms for information retrieval. In *International Conference on Image and Video Retrieval(CIVR)*, 2006.

[YH06b] R. Yan and A. G. Hauptmann. Probabilistic latent query analysis for combining multiple retrieval sources. In *Proceedings of the 29th international ACM SIGIR conference*, Seattle, WA, 2006.

[YH06c] Rong Yan and Alexander G. Hauptmann. Probabilistic latent query analysis for combining multiple retrieval sources. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 324–331, New York, NY, USA, 2006. ACM Press.

[YHJ03] R. Yan, A. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. In *Proc. of Intl Conf on Image and Video Retrieval*, pages 238–247, 2003.

[YLJH03] R. Yan, Y. Liu, R. Jin, and A. Hauptmann. On predicting rare class with SVM ensemble in scene classification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, 2003.

[YN05a] R. Yan and M. R. Naphade. Co-training non-robust classifiers for video semantic concept detection. In *IEEE Intl. Conf. on Image Processing(ICIP)*, 2005.

[YN05b] R. Yan and M. R. Naphade. Semi-supervised cross feature learning for semantic concept detection in video. In *IEEE Computer Vision and Pattern Recognition(CVPR)*, San Diego, US, 2005.

[YP97] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of the 14th ICML*, pages 412–420, 1997.

[YTFCD05] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *Proceedings of the 28th international ACM SIGIR conference*, pages 512–519, New York, NY, USA, 2005. ACM Press.

[YXW$^+$05] J. Yuan, L. Xiao, D. Wang, D. Ding, Y. Zuo, Z. Tong, X. Liu, S. Xu, W. Zheng, X. Li, Z. Si, J. Li, F. Lin, and B. Zhang. Tsinghua university at TRECVID 2005. In *NIST TRECVID 2005*, Nov 2005.

[YYH04] R. Yan, J. Yang, and A. G. Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 548–555, 2004.

[YZCJ02] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In *Proc. of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 688–693, 2002.

[ZCZ$^+$04] Y. Zhai, X. Chao, Y. Zhang, O. Javed, A. Yilmaza, F. Rafi, S. Ali, O. Alatas, S. Khan, and M. Shah. University of central florida at trecvid 2004. In *NIST TRECVID*, 2004.

[ZH01] X. S. Zhou and T. S. Huang. Comparing discriminating transformations and svm for learning during multimedia retrieval. In *Proc. of the ninth ACM international conference on Multimedia*, pages 137–146, 2001.

[ZL01] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of the 24th ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, 2001.

[ZL03] D. Zhang and W. S. Lee. Question classification using support vector machines. In *Proc. of the 26th ACM SIGIR*, pages 26–32. ACM Press, 2003.

[ZLC$^+$05] Y. Zhai, J. Liu, X. Cao, A. Basharat, A. Hakeem, S. Ali, M. Shah, C. Grana, and R. Cucchiara. Video understanding and content-based retrieval. In *NIST TRECVID*, 2005.

[ZR72] C. T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curve. *IEEE Transactions on Computers*, 21:269–281, 1972.