Representing multimedia documents by means of concepts – labels attached to parts of these documents – has great potential for improving retrieval performance. The reason is that concepts are independent from how users refer to them and from the modality in which they occur. For example, a Flower and une Fleur refers to the same concept and a singing bird can appear in an image or an audio recording. The question whether a concept occurs in a multimedia document is answered by a concept detector. However, as building concept detectors is difficult the current detection performance is low which causes the retrieval engine to be uncertain about the actual document representation.

This thesis proposes the Uncertain Document Representation Ranking Framework which deals with this uncertainty by transferring the principles of the Portfolio Selection Theory in finance – where the future win of a share is uncertain – to the concept-based retrieval problem. Similarly to the distribution of future wins, the retrieval framework considers multiple possible concept-based document representations for each document, which is the main scientific contribution of this thesis. In experiments for the shot and video segment retrieval task, the framework significantly improves performance over several baselines. Furthermore, simulations of improved concept detectors predict that concept-based retrieval will be suitable for large-scale real-life applications in the future.

ISBN: 978-90-365-3053-8
DOI: 10.3990./1.9789036530538
ISSN: 1381-3617, NO. 10-169

ROBIN ALY

Modeling Document Representation Uncertainty in Concept-Based Multimedia Retrieval

Modeling Document Representation Uncertainty in Concept-Based Multimedia Retrieval

ROBIN ALY

Invitation to the public defense of my thesis

Modeling Document Representation Uncertainty in Concept-Based Multimedia Retrieval

FRIDAY
JULY 2ND 2010
16:45

(INTRODUCTORY
TALK BEGINS 16:30)

ROOM WA 4
BUILDING WAAIER
(NO 12)

DRIENERLOLAAN 5
7522 NB ENSCHEDE

ROBIN ALY
ROBIN@ALY.DE

Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval

Robin Aly

PhD dissertation committee:

Chairman and Secretary:
    Prof. dr. ir. A. J. Mouthaan, Universiteit Twente, NL
Promotors:
    Prof. dr. P. M. G. Apers, Universiteit Twente, NL
    Prof. dr. F. M. G. de Jong, Universiteit Twente, NL
Assistant-promotor:
    Dr. ir. D. Hiemstra, Universiteit Twente, NL
Field expert:
    Dr. ir. R. J. F. Ordelman, Sound and Vision, NL
Members:
    Prof. dr. T. W. C. Huibers, Universiteit Twente, NL
    Prof. dr. C. H. Slump, Universiteit Twente, NL
    Prof. dr. W. Kraaij, Radboud Unversiteit Nijmegen/TNO, NL
    Dr. A. G. Hauptmann, Carnegie Melon University, USA

CTIT Dissertation Series No. 10-169
Center for Telematics and Information Technology (CTIT)
P.O. Box 217 – 7500AE Enschede – the Netherlands
ISSN: 1381-3617

SIKS Dissertation Series No. 2010-33
The research reported in this thesis was carried out
under the auspices of SIKS, the Dutch Research School
for Information and Knowledge Systems.

Printed by: Ipskamp Drukkers, Enschede, The Netherlands

ISBN: 978-90-365-3053-8
ISSN: 1381-3617, No. 10-169
DOI: http://dx.doi.org/10.3990/1.9789036530538

MODELING REPRESENTATION UNCERTAINTY

in

CONCEPT-BASED MULTIMEDIA RETRIEVAL

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee
to be publicly defended
on Friday, July 2nd, 2010 at 16.45

by

Robin Benjamin Niko Aly
born on October 5th, 1978
in Freiburg im Breisgau, Germany

Promotor: Prof. dr. P. Apers

Promotor: Prof. dr. F. M. G. de Jong

Assistant promotor: Dr. ir. D. Hiemstra

to my family

also to a society which favors
privacy and personal freedom over state control and pseudo safety.

Preface

In July 2006, I started my PhD endeavour at the University of Twente in the Strategic Research Orientation NICE (SRO-NICE)[1]. The initial research idea was to create a virtual cooking environment, assisting the cook on questions such as "How do I blanch broccoli?". Although the overall topic perfectly suited my personal interests, there were too many different aspects which needed to be addressed. Therefore, after investigating other possible topics, I decided to focus on one particular part: retrieving parts of videos. I was completely free in my decisions of choosing a research topic, which I mainly owe to the Dutch taxpayer - to whom I want to say special thanks at this point.

Acknowledgments

First of all, I would like to thank my two promoters Franciska de Jong and Peter Apers for providing such an ideal work environment. I really appreciate all the freedom I have had through the past four years. I also want to thank my daily supervisor Djoerd Hiemstra, without whom this thesis would literally never have seen the light of day. Through our numerous discussions, I improved in selling my ideas and learned that there is no model which is simply correct or incorrect. Before he partially changed to the Dutch Institute for Sound and Vision, Roeland Ordelman was my second supervisor and I also want to thank him for the many things I learned from him.

Many thanks go to the other committee members who were willing to judge this work. Alex Hauptmann deserves a special thanks for coming all the way from the U.S. to attend my defense.

A special thanks goes to Alan Smeaton for hosting my three month stay at DCU in Dublin. Along with the whole Clairty group, I would like to thank Aiden Doherty for the fruitful input and collaboration. My work benefited a lot from the stay on the green island. Furthermore, I am very grateful to Cees Snoek and Yu-Gang Jiang as well as their colleagues for providing me with detector scores. Without their huge prior effort and willingness to share research results this work would not have been possible.

I am also thankful to Arjen de Vries who helped me shape my research ideas especially in the early phase of my PhD - talking me away from unpub-

[1] `http://www.ctit.utwente.nl/research/sro/nice/`

lishable topics ignoring my unstructured rattling about “new ideas”. Claudia Hauff, who fought herself through many drafts of my papers, also deserves special thanks at this point.

Many thanks go to the members of the DB group, who created such a nice working atmosphere and gave me a lot of support in various aspects of my work. I would like to distinctly mention Maurice van Keulen without whom I would not have come to the UT and Ida den Hamer-Mulder who is in my opinion the most important person in the operational business in the group. From the HMI group, I mainly want to thank Lynn Packwood for the last minute language proofreading of my thesis.

Last but not least, I would like to thank my family – including my Computer-Grandma Helena – and my friends. Without all the love and support I got from all of you I would not have made it.

Robin Aly
Enschede, July 2010

Contents

Chapter 1

Introduction

1.1 Why do we need Concept-Based Multimedia Retrieval?

More and more of our lives is captured in digital *multimedia documents*, such as audio recordings, pictures or videos. For example, many children have a digital second life in the form of thousands of photos and endless hours of video footage being captured from the very moment of their birth. On the other extreme, patients suffering from amnesia can be helped through their external memory, which is automatically recorded by a camera taking more than 2,000 photos per day (Berry et al., 2009). Furthermore, in the professional domain, multimedia documents are a necessity. For example, press agencies store digital images and videos of almost every event of public interest (Enser, 1995), and cultural heritage archives digitize their multimedia assets for preservation and improved accessibility (Heeren et al., 2009).

There are the following main explanations for this trend. First, since the mid-1990s the production and storage of new content as well as the digitization of existing content has become constantly easier and cheaper. Second, some information types, for example learning material, can be faster absorbed via multimedia documents than by text (Moreno and Mayer, 1999). Finally, for many people multimedia content is more attractive than text – "A picture is worth a thousand words". As a result, multimedia collections grow rapidly, both in terms of numbers and volume. This growth and the wealth of information in the collections make an automated search facility (called a *retrieval engine*), which fulfills a user's *information need*, indispensable. The research discipline aiming to improve this search is called multimedia retrieval and is derived from the more general field of information retrieval.

In order to find documents which fulfill an information need, retrieval engines base their search on document representations. Today, most multimedia retrieval engines use document representation of manually created, textual metadata, such as assigned keywords (tags) (Ames and Naaman, 2007). Ranking multimedia documents using textual document representations often returns good results, since well performing text retrieval engines can be re-used. However, the use of manually created metadata also has serious limitations. First, the metadata is time consuming to create. Second, if the metadata is created by laymen it is subjective and ad-hoc ("How did I name this picture again?") and employing professionals to create metadata is expensive (Ordelman et al., 2007). Finally, because of the amount of required metadata it is practically infeasible to allow users to search for particular segments inside a video.

Concept-based multimedia retrieval which is based on document representations consisting of automatically detected *concept* occurrences was proposed to improve upon the limitations of manually created metadata, see Naphade and Smith (2004) for an overview of this emerging research discipline. For this introduction, the reader can think of a concept as a label attached to a (part of a) multimedia document where all users agree that this label is appropriate. For example, a concept could be a *Flower*, a *Car* or a scene being *Outdoor*. Here, we refer to concepts by English terms. However, these terms are just references to the concept which itself is language-independent and could be referred to in other languages or by computer codes. For example, the concept *Flower* could also be referred to as *Fleur* (French for Flower) or #F1 (a reference to this concept in a computer). Furthermore, a concept is modality independent[1]. For example, the concept *Singing Bird* can occur in the visual modality as well as in the audio modality[2]. Note that there are other research areas in information retrieval which use concepts, for example in the biomedical domain (Trieschnigg et al., 2009) or for the description of web pages (Loh et al., 2000). However, in this work we will focus on the use of concepts in multimedia retrieval.

The main advantages of concept-based multimedia retrieval are the following. First, the detection of concepts is performed by computers and is therefore cheaper and less time consuming to perform than manual creation of metadata. Second, a retrieval engine relying on textual metadata will have problems fulfilling a users' information need corresponding to the animal *Jaguar* when he expresses this need by the term 'Jaguar' in the query. The reason is that the retrieval engine cannot determine whether metadata which contains the term 'Jaguar' refers to an animal or to a car. However, in concept-based retrieval this is not a problem, once the retrieval engine knows that the user is referring to the animal concept *Jaguar*. Finally, the modality independence of concepts simplifies the unified retrieval of different kinds of multimedia documents. For example, searching for "Singing Birds" can return images of a singing bird or audio recordings.

Unfortunately, concept-based retrieval is not yet ready for large-scale application in the real world. The main obstacles are the following. First, it is difficult to automatically detect concepts in multimedia documents (Yang and Hauptmann, 2008a) since the appearance of concepts is often different.

[1] A modality is a sense through which the human can receive the output of the computer.

[2] A more elaborate and precise definition of concepts can be found in Chapter 2.

For example, cars exist in many different colors and shapes making it difficult for a computer to detect that all of them are *Cars*. Second, translating the user's information need into concepts is problematic, since a retrieval engine has to find, for instance, the correspondence between the concept *Jaguar* and the way a Chinese user would express this information need (Natsev et al., 2007). Finally, current research is concentrated on searching for short bits of videos. For example, fulfilling the information need "Find me a jaguar". Here, a suitable document representation is the occurrence of a *Jaguar*. However, users can also be interested in longer video segments (Vries et al., 2004), for example for the information need "Find me hunting jaguars". Here, representing a video segment by the occurrence of a *Jaguar* is not expressive enough. The retrieval engine cannot differentiate between segments where a *Jaguar* is only briefly shown and segments to which the concept is important. Therefore, the document representation should contain the importance of a *Jaguar* in a video segment to fulfill the information need.

The remainder of this chapter is structured as follows. Section 1.2 introduces the basic components of a retrieval engine. Section 1.3 identifies the main problems in multimedia retrieval addressed in this thesis. In the following section, Section 1.4 defines the scope of this thesis and gives an overview of the proposed approach. Then, Section 1.5 explains the research questions. In Section 1.6, an outline of the remainder of this thesis is presented.

1.2 The Basic Components of a Retrieval Engine

This section introduces the basic components which are commonly used by retrieval engines. This basic vocabulary is introduced because this thesis adds to it, see Section 1.4. The basic components of a retrieval engine are motivated by the root challenge of information retrieval which is described by Spärck-Jones and Willett (1997) as follows.

> "The root challenge in retrieval is that (information-) user need and document content are both unobservable, and so is the relevance relation between them".

Figure 1.1 shows the basic components of a retrieval engine inspired by the conceptual model for information retrieval by Fuhr (1992) and the following discusses the components shown.

The three topmost components in Figure 1.1, information need, document content and relevance are the central objects in information retrieval. They are, according to Spärck-Jones and Willett (1997), unobservable, which means that the computer cannot comprehend their meaning, which is actually a question of not being able to represent their content. For example, a retrieval engine will never be able to capture all aspects of a painting by van Gogh or an information need corresponding to "exciting times", certainly

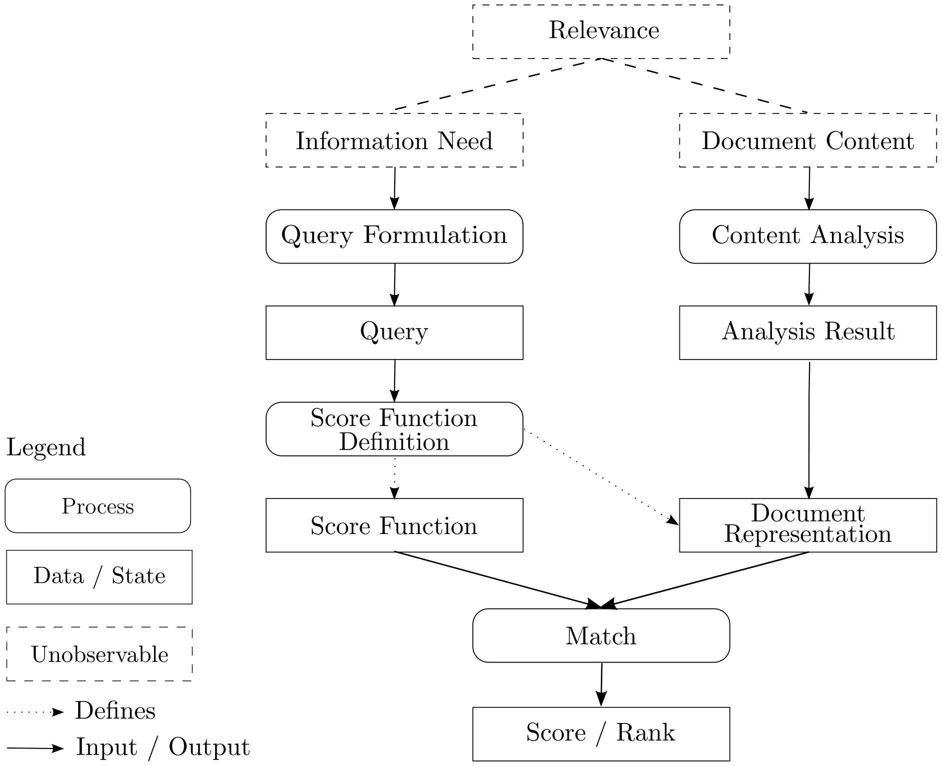


Figure 1.1: *The basic components of a retrieval engine based on the conceptual model for information retrieval by Fuhr (1992).*

because they will differ from user to user. As a result, the relevance of a document to an information need is also unobservable.

In order to represent a document, the *content analysis process* extracts features from each document (see the right part of Figure 1.1). The output of the content analysis process is called the *analysis result* and consists of all supplied *features* produced by the content analysis process.

During the *query formulation* process an information need is translated by a user into a *query*, which can be processed by the retrieval engine (see the left part of Figure 1.1). Based on a query, the *score function definition* process performs three sub-processes.

(1) The score function definition determines the *document representation*, which will be used to answer the query, by selecting a subset of the features of the analysis result. For a concept-based retrieval system, this sub-process selects the concepts which should be used to answer a query.

(2) The score function definition estimates a weight for each selected feature in the document representation.

(3) The score function definition defines a *score function* which takes document representations as arguments and uses the estimated weights to calculate a ranking score value.

In this work the first two sub-processes are jointly referred to as the *concept selection and weighting* process.

A *retrieval model* is the theory behind the score function definition process and is not shown in Figure 1.1. In text retrieval the research of retrieval models has received considerable attention, which is one of the reasons for the success of internet retrieval engines today (Baeza-Yates and Ribeiro-Neto, 1999). On the other hand, in concept-based multimedia retrieval, retrieval models do not receive as much attention because the content analysis process is perceived as the biggest bottleneck to performance (Snoek and Worring, 2009).

The *match* process iterates over all documents of a collection and applies the score function to the document representation, resulting in a ranking score value for each document. The documents are then sorted in descending order by the ranking score value to produce the answer to the query, a ranked list of documents.

Figure 1.1 has the following differences with the conceptual model for information retrieval by Fuhr (1992). First, the mathematical symbols for the components, used by Fuhr (1992), replaced in Figure 1.1 by descriptive text. Second, the score function definition process has been used instead of two processes by Fuhr (1992), one which defines the features belonging to the document representation and one which separately defines the weights of these features. A single process was chosen because the results of both alternatives – Fuhr's and ours – have to correspond (the weightings have to match the features in the document representation). Furthermore, in our definition, for each query a new score function is defined and invoked in the matching process while Fuhr (1992) defines the score function as an anonymous part of the matching process. We opt for an explicit definition of the score function because of its importance in later chapters.

1.3 Fundamental Problems in Content-Based Multimedia Retrieval

Content-based multimedia retrieval, of which concept-based multimedia retrieval is a relatively new sub-discipline, has the benefit that it does not depend on manually created metadata because it relies on document representations which are created by a purely computer-based content analysis process. However, the problems of content-based multimedia retrieval can be demonstrated on the basis of the two most active sub-disciplines.

In *content-based image retrieval* document are typically represented by high dimensional vectors, called low-level features, which are only interpretable by computers. The problem of representing documents by their low-level features is also often referred to as the semantic gap (Smeulders et al., 2000):

> "The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation. [...] A user looks for images containing certain objects or conveying a certain message. Image descriptions, on the other hand, rely on data-driven features and the two may be disconnected."

Here, the data-driven features refer to the document representation consisting of low-level features in our terminology. Therefore, it is impossible to directly match low-level features onto information needs, the query formulation process that is best understood is the so-called query-by-example paradigm where a user has to produce an example picture which is used for retrieval. However, Markkula and Sormunen (2000); Rodden et al. (2001) show that it is difficult for a user to formulate his query using this paradigm. Furthermore, when content-based image retrieval techniques are adapted to *video retrieval*, low-level features are extracted for discrete time units, for example *video shots* of around ten seconds length[3]. However, if information needs refer to longer video segments the features which could be used to rank documents can be spread over the whole video segment. This makes the score function definition more difficult because the document representations of low-level features now also contain a time dimension which is difficult to include in a score function.

On the other hand, in *spoken document retrieval* transcripts of the spoken words are used as a document representation. Spoken document retrieval produces poor retrieval results if the transcript contains errors, for instance because the recordings were taken in noisy surroundings and important words were not included in the transcript (Mamou et al., 2006). Additionally, to predict whether retrieval engines will perform better if the number of errors reduces is a problem frequently addressed in content-based retrieval (Witbrock and Hauptmann, 1997). Furthermore, in spoken document retrieval, events, such as *Applause*, are normally not included in the transcripts and therefore the search is limited to the spoken content.

In concept-based retrieval, *concept detectors* try to detect the occurrence of a predefined set of concepts (the *concept vocabulary*). The research on concept detector techniques is mainly focused on video data (Snoek and Worring, 2009) but was also proposed for image data (Wang et al., 2008) and audio data (Lu, 2001; Peng et al., 2009). After the detection process, which is performed off-line, a retrieval engine uses the detector output to answer queries.

In theory, there are several advantages of concept-based retrieval over other content-based retrieval methods. First, the query formulation in concept-based retrieval is improved compared to the one in content-based image retrieval. The reason is that with a sufficiently large concept vocabulary most queries can be expressed by concepts, which is difficult with low-level features

[3] See Section 2.3.1 for a definition.

because of the semantic gap. Second, compared to spoken document retrieval engines which limit the document representation to transcripts of the spoken content, concept-based retrieval can represent events, such as *Applause*.

However, although some problems are reduced in concept-based retrieval, the following problems persist and will be addressed by this thesis.

P1 **Document Representation Uncertainty** Today, the performance of concept detectors is often limited. As a result, the detectors often wrongly decide whether a concept occurs in a video or not. This leads to *document representation uncertainty* which is the reason why most current approaches use the detectors' confidence about the concept occurrence as a document representation instead of the actual occurrence. However, a major problem with using this document representation is that score functions are difficult to define (Snoek and Worring, 2009) and the search performance is limited (Yang and Hauptmann, 2008a).

P2 **Query Formulation Support** Users may not always be familiar with the, possibly large, concept vocabulary of the retrieval engine. Furthermore, the definition of score functions requires concept-specific weights which often depend on the collection, which the user is normally not familiar with. Therefore, it is difficult for users to formulate queries by selecting concepts themselves and a user interface has to support the user in formulating his query.

P3 **Support for Longer Video Segments** Concept detection is usually done on a video shot level. However, users can also be interested in finding longer video segments (Vries et al., 2004) and the occurrence of useful concepts can be spread over the shots of the longer video segment. It is therefore a problem how to combine the detector output of multiple shots for retrieval. This problem has been pointed out occasionally, but up till now was under addressed.

P4 **Search Performance Prediction** As mentioned before, concept detector performance is currently still limited, which leads to limited search performance. Therefore, the prediction of the search performance of current retrieval engines under improved detector performance is an important problem for justifying the research effort put into concept-based retrieval (Hauptmann et al., 2007).

1.4 Scope and Overview of the Proposed Approach

This thesis considers pure concept-based retrieval without query refinement after the initial query. Note that this sometimes leads to a lower performance compared to combining modalities (for example, concepts with text) and

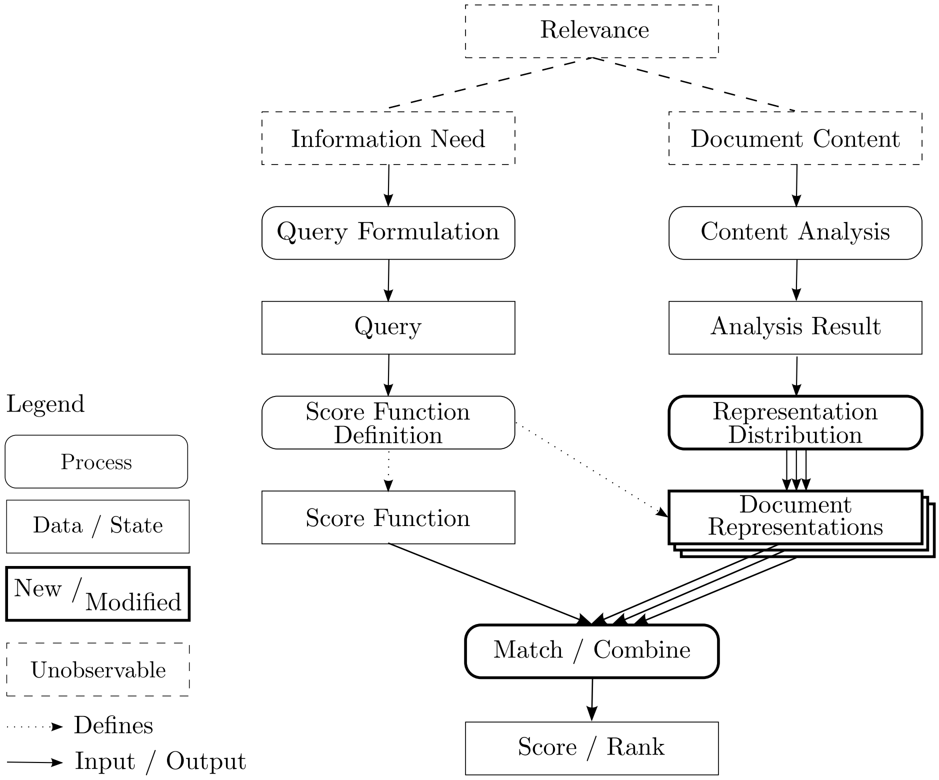


Figure 1.2: *Changes in the components of a retrieval engine proposed by this thesis.*

including user interaction (Snoek and Worring, 2009; Yan, 2006). However, there are the following advantages of this focused scope:

- The focused scope allows an isolated investigation of the effects of document representation uncertainty for concept-based document representations.
- The proposed techniques are applicable to collections where information from some modalities is not available. For example, pure concept-based search can be used for the application area of surveillance cameras where no spoken text is extracted.
- The techniques are also applicable if other modalities are available and if user interaction is allowed.

This thesis describes the principal ingredients to address the fundamental problems of concept-based retrieval described in Section 1.3. The main theoretical contribution of this thesis is the Uncertain Representation Ranking (URR) framework which is derived from the Nobel Price winning Portfolio Selection Theory by Markowitz (1952). The URR framework explicitly models the document representation uncertainty by allowing multiple document representations per document. The framework replaces the classical retrieval process, shown in Figure 1.1, by the one shown in Figure 1.2.

In Figure 1.2, the features of the analysis result (the detector output) are not directly used as a document representation. Instead, the possible document representations consisting of concept occurrences and absences, called a concept-based document representation, are used for ranking. Based on the analysis result, the new *representation distribution* process assigns each concept-based document representation a probability of being the actual representation. In the *match* process, a retrieval score value is calculated for each document representation possibly re-using an existing text retrieval model. Afterwards, a new *combination* process combines the retrieval score values of the possible document representations to a final ranking score value of the document based on the previously defined probabilities. The described changes of the retrieval components have the following advantages.

Re-use of Text Retrieval Models Concept-based document representations are similar to existing document representations in text retrieval. For example, the occurrence of a concept in a multimedia document can be compared with the assignment of an index term to a book, a document representation which has often been used in text retrieval, see for example Maron and Kuhns (1960); Robertson et al. (1982). Therefore, text retrieval models can be re-used for concept-based retrieval. This improves the score function definition process because of the following reasons. First, the mathematical blueprint (Hiemstra, 2001) of a score function in a text retrieval model has proven to be successful (Baeza-Yates and Ribeiro-Neto, 1999). Second, it is easier to set weights for a concept occurrence than for a detector's confidence. For example, collection statistics similar to the well-known inverse document term frequency (Spärck-Jones, 1972) express the importance of a concept which can be used for the assignment of weights.

Longer Video Segments If longer video segments are considered as a series of video shots, the concept occurrences of the shots in a segment can be combined to the concept frequency of the segment. Furthermore, concept frequencies correspond to some extent with term frequencies. This is the case since both are intuitively a measure of the importance of the concept or term in a document – the more frequent a *Jaguar* occurs in a video segment, the more important the concept is to this segment. Term frequencies and inverse document frequencies are used in many existing text retrieval models today which allows re-use of these models for concept-based retrieval for longer video segments, which is problematic with current multimedia retrieval approaches.

1.5 Research Questions

The following research questions, which can be derived from the stated problems in Section 1.3, are answered by this thesis:

Q1 *Can a general framework be defined for document representation uncertainty, which re-uses text retrieval for concept-based retrieval?*

Q2 *How can the document representation and its weights be defined automatically and in a user-friendly manner for an information need?*

Q3 *How can the retrieval of longer video segments be supported based on concept occurrence in video shots?*

Q4 *What is the impact of the proposed ranking framework and the concept selection and weighting method on the retrieval performance?*

Q5 *How can we predict whether improved concept detection will make a current concept-based retrieval engine applicable to real-life applications in the future?*

1.6 Outline

This section describes the structure of this thesis.

Chapter 2 describes work related to this thesis. First, basic notation and definitions are given. Second, the basics of concept detection techniques are described, which are needed to understand this thesis. Finally, existing retrieval models for concept-based multimedia retrieval are reviewed by checking whether they contain proposed desirable properties.

Chapter 3 presents the URR framework, a general framework for ranking documents, based on multiple possible document representations and combining the resulting scores into a single retrieval score value. The framework transfers the Portfolio Selection Theory in finance by Markowitz (1952) to the problem of ranking documents with uncertain document representations. This chapter emerged from the ideas presented in Aly (2009).

Chapter 4 describes a method to select a concept-based document representation and set the concepts' weights for an information need. The proposed method uses a development collection, created to train concept detectors, for which a textual representation is created. For a textual query, a text retrieval engine ranks the development collection and this ranking and the known concept occurrences are used to select the concepts and set required weights. This chapter is based on Aly et al. (2009) which emerged from our joint work (Hauff et al., 2007).

Chapter 5 applies the URR framework from Chapter 3 to video shot retrieval, using the probability of relevance retrieval model (Robertson, 1977), based on a document representation of binary concept occurrences. This chapter is based on Aly et al. (2008a) and was evaluated in the TRECVid evaluation campaign in Aly et al. (2008b).

Chapter 6 applies the URR framework from Chapter 3 to the retrieval of longer video segments. Here, concept frequencies are used as a document representation. By using their similarity to term frequencies, the language model ranking function from text retrieval (Hiemstra, 2001) is adapted to concept-based retrieval. This chapter is based on Aly et al. (2010).

Chapter 7 proposes a method to simulate concept detectors and the simulation result is used to show that the concept-based retrieval paradigm can show good results. This chapter is based on Aly and Hiemstra (2009a) and accompanying material provided in Aly and Hiemstra (2009b).

Finally, Chapter 8 draws conclusions from the answers to the research questions given in this thesis and proposes future work.

Chapter 2

Concept-Based Retrieval Models

2.1 Introduction

As mentioned in Section 1.2, retrieval models are treated less formally in concept-based retrieval than in text retrieval. This can be seen from the fact that in most works the retrieval function is pragmatically described as a weighted sum, where weights and summands often do not carry a thoroughly defined meaning (Kennedy et al., 2008; Snoek and Worring, 2009).

In the following, a review of existing state-of-the-art retrieval models is presented. The aim of the review is to investigate the way the retrieval models attempt to solve the problems P1 and P2 described in Chapter 1:

P1 **Document Representation Uncertainty** Today, the performance of concept detectors is often limited. As a result, the detectors often wrongly decide whether a concept occurs in a video or not. This leads to *document representation uncertainty* which is the reason why most current approaches use the detectors' confidence about the concept occurrence as a document representation instead of the actual occurrence. However, a major problem with using this document representation is that score functions are difficult to define (Snoek and Worring, 2009) and the search performance is limited (Yang and Hauptmann, 2008a).

P2 **Query Formulation Support** Users may not always be familiar with the, possibly large, concept vocabulary of the retrieval engine. Furthermore, the definition of score functions requires concept-specific weights which often depend on the collection, which the user is normally not familiar with. Therefore, it is difficult for users to formulate queries by selecting concepts themselves and a user interface has to support the user with formulating his query.

As there is currently no concept-based retrieval model which addresses problem P3, the support for retrieval of longer video segments, it is left out from the review, and the reader is referred to Chapter 6 where a method to rank longer video segments is proposed. Furthermore, the problem P4, the prediction of the search performance of current retrieval engines under improved detector performance, is discussed in Chapter 7 since it is not a requirement of an operational retrieval engine. For an overview over the rest of this chapter, Figure 2.1 shows an example of the components of a video retrieval engine with references to the sections where the individual content is discussed.

The remainder of this chapter is structured as follows: In Section 2.2, notation and the basic definitions are introduced. Section 2.3 gives a brief overview of current content analysis (especially concept detection) techniques, since they have strong impact on the performance of retrieval. Afterwards, in Section 2.4, state-of-the-art concept-based retrieval functions are reviewed and evaluated. Section 2.5 evaluates selection and weighting methods which select the features for a document representation and assign weights to these features. Finally, Section 2.6 summarizes this chapter and discusses the results.

2.2 Notation, Definitions and Evaluation Test Bed

This section introduces the basic notation used in this thesis. Afterwards, the notion of concept and information need and their relations are defined. These are the most central notions in this thesis. Finally, the section is ended by describing the TRECVid workshop, which is used as an evaluation platform throughout this thesis.

2.2.1 Basic Notation and Terminology

In this section, the notation which is used in this thesis is introduced. Note that a condensed overview of the notations and definitions in this thesis is provided on page 175.

The central objects in information retrieval are defined by: let d be the current document and $\mathcal{D} = \{d_1, ..., d_N\}$ the current search collection. Furthermore, let Ω be the "universe of documents" which will be defined in Section 2.3. The current information need is denoted by *infneed* and the query, in which a user expressed the need *infneed*, is denoted by q. As commonly done in the literature, a query is modeled as a document.

Features and Document Representations In the following, the notation for the query and document representations is presented. This thesis differentiates between features (the color of *a* car) and feature values (the color of *this* car is red): a *feature* F is a function from a document to a value in the feature domain $dom(F)$ of this feature, $F : \Omega \to dom(F)$[1]. On

[1] Strictly speaking, $dom(F)$ is the range of the feature function. However, $dom(F)$ is also used the in context of random variables where the term domain is often used.

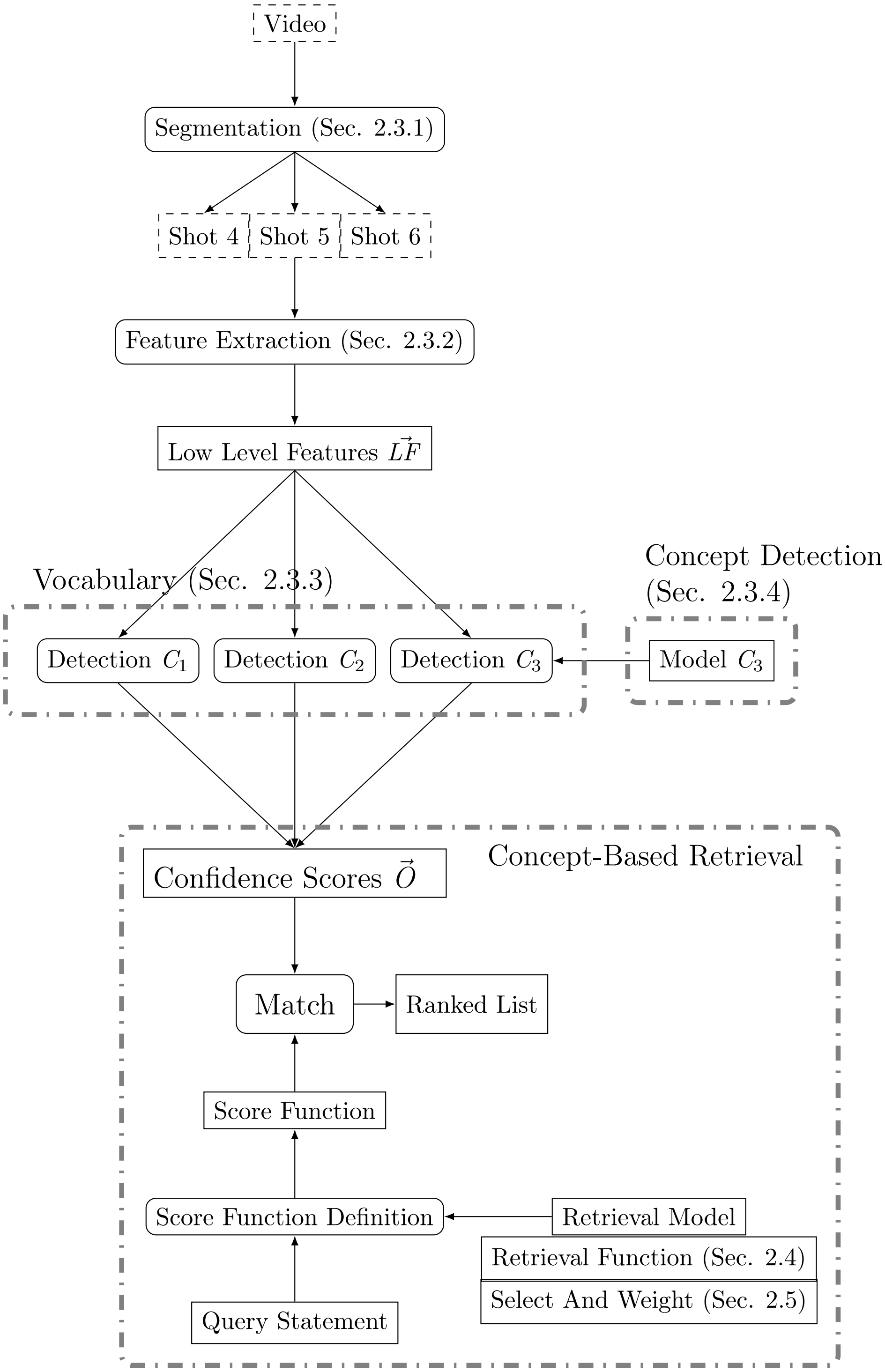

Video
Segmentation (Sec. 2.3.1)
Shot 4
Shot 5
Shot 6
Feature Extraction (Sec. 2.3.2)
Low Level Features $\vec{LF}$
Vocabulary (Sec. 2.3.3)
Concept Detection (Sec. 2.3.4)
Detection C_1
Detection C_2
Detection C_3
Model C_3
Confidence Scores $\vec{O}$
Concept-Based Retrieval
Match
Ranked List
Score Function
Score Function Definition
Retrieval Model
Retrieval Function (Sec. 2.4)
Select And Weight (Sec. 2.5)
Query Statement

Figure 2.1: *Detailed process diagram of a concept-based retrieval engine.*

the other hand, a *feature value* results from the application of the feature function to a document d. To keep features and feature values separate, we use a smaller case letter for the function when it is applied to a document. Therefore, the feature value of the feature F for document d is $f(d)$. To improve the readability of the notation, in unambiguous cases, we drop the argument d from a feature value notation when the variable refers to the current document. The document feature vocabulary $\mathcal{V}$ is the set of features, which are provided by the content analysis process. For example, a possible feature is the concept detector output, a so-called confidence score, denoted by O, see Section 2.3.4 for a definition. The confidence score for a document d is then denoted by $o(d)$ (or o in unambiguous cases), and the vocabulary $\mathcal{V}$ is a set of all confidence score features for which concept detectors exist, $\mathcal{V} = \{O_1, \ldots, O_{|\mathcal{V}|}\}$. Features and feature values are also addressed by identifiers. For example $F_{US-Flag}$ is used to refer to the feature concerning the concept US-Flag.

Let the *document representation* be a vector of features $\vec{F} = (F_1, \ldots, F_n)$ which are used for the current query. Similarly, let the values of a document representation for a document d be the vector of feature values $\vec{f}(d) = (f_1(d), \ldots, f_n(d))$ (or only $\vec{f}$). The set of all possible representations of $\vec{F}$ is denoted by $dom(\vec{F}) = dom(F_1) \times \ldots \times dom(F_n)$.

Since queries are modeled as documents, they also have features. However, the features used for queries are not necessarily the same features as the ones used for documents. Since the focus of this thesis is document features only a single query feature set is introduced. The same notation of feature vectors and feature value vectors of the document representations is used. Unless stated otherwise, all queries features in this thesis will be term frequency features, $\vec{QF} = (TF_1, \ldots, TF_{|\mathcal{TV}|})$, where each feature TF_i counts the occurrences in a document of the ith term in a list of terms $\mathcal{TV}$ from a language.

Retrieval Models A *retrieval model* is the theory behind the score function definition process, see Section 1.2 and consists of two components.

Selection and weighting the *selection and weighting* procedure which defines how to arrive from a query at a document representation and what weights to assign to each feature of the representation.

Retrieval Function The retrieval function is a blueprint of a score function (Hiemstra, 2001).

The notation of the components of a retrieval model is defined in the following. In order to identify a certain retrieval model an identifier in the subscript is used. For the purpose of this definition, we use the generic identifier M. The selection and weighting is performed by the function $selectNweight_M$ which takes query feature values as arguments and returns the document representation (the features), $\vec{F}$, and a query-specific weighting function $w : \mathcal{V} \rightarrow \mathbb{R}$, which maps features to weights. Since w is always used in correspondence with the feature vector $\vec{F}$, the weight of the ith feature is also sometimes denoted by w_i, meaning $w(F_i)$ where F_i is the ith component of $\vec{F}$.

A retrieval function is denoted by $\mathit{retfunc}_M\langle\vec{F}, w\rangle$. Since the retrieval function $\mathit{retfunc}_M\langle\vec{F}, w\rangle$ is a template of score functions and the <> operator denotes the use of template parameters, as known from modern programming languages (Stroustrup, 2000). The retrieval function is not a function in a strict mathematical sense since the arguments are not fixed yet (queries can have different document representation). For example, the retrieval function weighted sum might be defined as follows.

$$\mathit{retfunc}_{\mathit{CombSUM}}\langle\vec{O}, w\rangle(\vec{o} : \mathit{dom}(\vec{O})) = \sum_i w(O_i)\ o_i$$

Here, the retrieval function is defined on an arbitrary set of features with a corresponding weighting function. The calculation of the score is defined for an arbitrary set of feature values $\vec{o}$ (the document representation of particular document) and is calculated as the sum of the feature values weighted by the feature weight $w(O_i)$. However, it is not yet defined, what the template parameters $\vec{O}$ and w are. For a particular query q, the score function score_q is a query-specific instance of the retrieval function where the derivation is denoted by $\mathit{score}_q := \mathit{new}\ \mathit{retfunc}_M\langle\vec{F}, w\rangle$. Here, $\vec{F}$ is a document representation selected for the query and w is the corresponding weighting and score_q is a specific weighting function defined on $\vec{F}$.

Probabilities This thesis makes frequent use of probability theory. Therefore, the most essential notions from this theory are defined here. Probabilities are always used in reference to a probabilistic event space (a set of events). Throughout this thesis document events are considered and the uniform probability measure is used, which assigns all events the same probability. The event space will be denoted as a subscript of the probability measure, for example P_Ω is the probability measure on the event space of the document universe Ω. The standard probability measure, denoted by $P()$, is the probability measure with the event space of the documents in the collection $\mathcal{D}$. Furthermore, random variables are functions from events to the function's range, which is called domain, by convention. They will be denoted in upper case. Note, that considering the event space of document events, the definition of a feature and a random variable are equivalent. Besides this essential notation, a more elaborate description of the aspects of probability theory, which are used in thesis, can be found in Appendix A.

2.2.2 Concepts, Information Needs and Relevance

This section provides definitions of the two most central notions in this thesis, a concept and an information need. First, a definition of a (semantic-)

concept is developed. Similar to Snoek and Worring (2009), the definition is based on Aristotle's work on categories. The work identifies ten different atom categories, which cannot be split any further. They are: Substance, Quantity, Quality, Relation, Place, Time, Position, Action and Affection. The most central category is the Substance. In contrast to the definition from Snoek and Worring (2009), this thesis distinguishes a category and a concept by the definition of a substance concept, see Millikan (2000):

> "Substance concepts are primarily things we use to think with rather than talk with. [...] Having a substance concept is having a certain kind of ability - in part, an ability to reidentify a substance correctly [...]".

Therefore, the main difference between categories and concepts is that a concept is a mental representation of a category, which allows us to reidentify the category. This indirection of a concept as a mental representation is introduced, because it is hard to imagine abstract categories to be contained *in* a video. For example, everybody has a concept *Outdoor* which is used to reidentify the underlying category. A cartoon does not show the category *Outdoor* although most people will reidentify the category, if the sketched scene shows the sky and so forth. Therefore, in the definition of this thesis it is only of importance whether the human reidentifies a category (by his concept), not whether the category is actually present.

The following assumptions about concepts are made. First, a concept is always fully present or not present at all, never a bit. Second, the occurrence of a concept has a universal truth, meaning that it is always objectively identifiable whether a concept occurs. This thesis uses the words *concept occurs*, to denote that a person can reidentify a category through a concept in a document and the *concept is absent*, if it cannot be reidentified. When a user explicitly states that a concept occurs in a document, he *annotates* the document with the concept occurrence.

Definition 2.1. Let $C : \Omega \to \mathbb{B}$ be the concept occurrence feature. The concept occurrence feature value of a concept C in a document d is defined as follows:

$$c(d) = \begin{cases} 1 & \text{if the concept occurs in } d \text{ ;} \\ 0 & \text{otherwise.} \end{cases}$$

Information Needs Taylor (1962) was the first to characterize the query formulation process starting from an information need. In his original work the process consists of four stages where the last two are called query formulation and query. The first two stages define what is referred to as an information need in this thesis:

> "(1) The conscious or unconscious need for information not existing in the remembered experience of the investigator. [...] (2) In progressing toward the concrete, the next form of need is the conscious mental description of an ill-defined area of indecision".

This definition of an information need has similarities to that of a concept. However, this thesis distinguishes the two: an information need is normally more complex in structure and – more importantly – the relevance to an information need is subjective in nature; in contrast to concept occurrences which are assumed to have a universal-truth. That is, although the relevance to an information need is sometimes specified in the same way as concept occurrences, for example by several relevance judges, we assume that it is an individual who poses the query to a retrieval engine and his idea of relevance between a document and his information need might or might not coincide with the one from other users with an equivalent query. In the following the relevance relation between a document and the current information need is defined.

Definition 2.2. Let $R : \Omega \rightarrow \mathbb{B}$ be the relevance relation between a document and the current information need, which is defined as follows:

$$r(d) = \begin{cases} 1 & \text{if the document } d \text{ is relevant to the current information need;} \\ 0 & \text{otherwise.} \end{cases}$$

Note, this definition is equivalent to the one in the well-known probability of relevance ranking principle in information retrieval, see Robertson (1977). Since this thesis never considers more than one information need at a time, the notation does not emphasize that $r(d)$ and R are always related to a particular information need *infneed* – which is not the case in some other retrieval models described below, where it will be explicitly mentioned.

Parallel between Concepts and Index Terms To establish a link to text information retrieval, which will be used in Chapter 5, and further below in this chapter, we give the definition of an index term in library science. The definition of an index term can be derived from the definition of the coordinate indexing process described by Taylor (1962).

> "The enabling of information retrieval through the use of related terms in a catalog or database to identify concepts".

Therefore, a librarian decides whether a document should be indexed with a certain term, possibly including synonyms, or not which is then assumed to be universally-true. As a consequence, index terms can be thought of nearly being equivalent to concepts.

Definition 2.3. Let $T : \Omega \rightarrow \mathbb{B}$ be the feature which yields whether a document is indexed under a certain term.

2.2.3 TRECVid: An Evaluation Campaign

Comparability of results is an important topic in many research disciplines. This has two reasons. First, it is difficult to obtain comparable datasets (for example hindered by copy rights). Second, without a central standardization body, different evaluation measures would be used; also hindering comparability. The annual TRECVid workshop, organized by the National Institute for Standards and Technology (NIST), has the aim to tackle this problem by providing standardized collections and evaluation measures (Smeaton et al., 2006). The evaluation of the methods proposed in this thesis is based on the data provided by this workshop. Therefore, the most relevant aspects of the workshop are described in the following.

Collections and Information Needs Every year, the workshop organizers provide the participants with a video (search) collection which is segmented into shots by a common shot boundary definition, see Section 2.3.1 for further explanation. Additionally, for the training of concept detectors and retrieval engines, a training collection from the same domain is provided. In the years from 2002 until 2006 the domain of the videos was broadcast news. Later, in 2007 until 2009 data from the Dutch Institute for Sound and Vision, containing general Dutch television, were used. This thesis contains experiments using the collections from workshop years 2005-2009.

The information needs describing the search tasks for the workshop participants are formulated through a set of sample images or video clips and query texts. Because example images are also documents, the syntax $q.s_i$ is used to specify the ith example image or video of the query q. Furthermore, compared to the average 2.5 words which users employ in current web search engines, the query texts are long with 8.8 terms on average plus a common prefix of “Find shots of..” in the used collections. Furthermore, the query texts have a relatively regular structure.

Tasks for Participants There are multiple tasks in which participants of the TRECVid workshop can participate. However, only two of them are of importance in this thesis: the *high-level feature extraction task* and the *automatic search task*. In the high-level feature extraction task the workshop participants have to return for each concept from a list a ranked list of shots. The list should be sorted in decreasing likelihood that the concepts occur in each shot. The output of detectors of this task will be used by the retrieval models proposed in this thesis. In order to train concept detectors the research community collaboratively annotates the training collection. In the automatic search task, a set of queries has to be fully automatically processed and a ranked list of shots has to be returned. This is the task which is approached in this thesis.

Evaluation Measures Both the high-level feature extraction and the automatic search task are evaluated using the mean average precision (MAP). The measure MAP is based on relevance judgments and concept annotations on the search collection. Since complete relevance judgments and concept

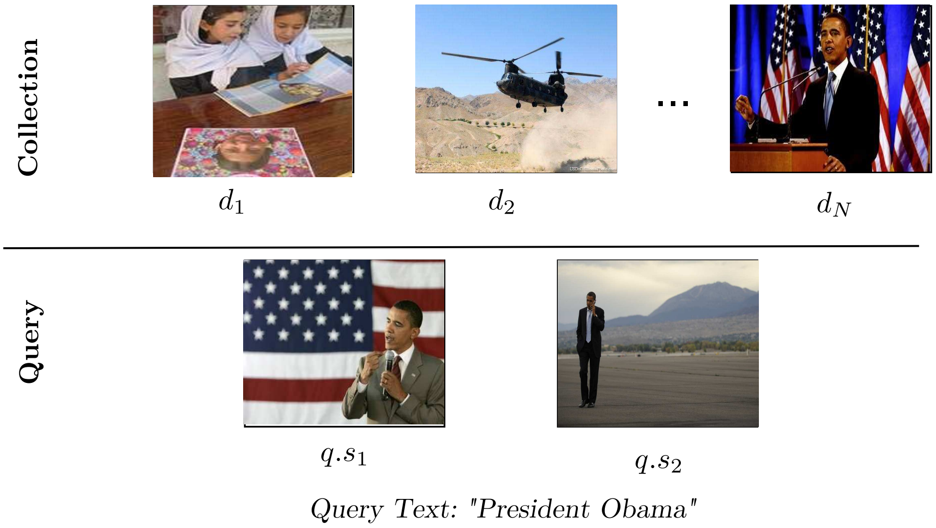


Figure 2.2: *An example of a collection and a query.*

occurrence annotations are not feasible, the set of relevant documents is determined from a pool of the first 100 returned shots by the participants, which is a procedure also used in the wider known text retrieval evaluation campaign (TREC) workshop from which the TRECVid workshop originated (Harman, 1995). In order to further reduce the costs of relevance judgments and concept occurrence annotations while still allowing a sufficient pool depth the workshop organizers introduced a new evaluation method in 2006 where only randomly selected documents from the pool are judged and the inferred mean average precision (infAP) instead of the mean average precision is calculated, see Yilmaz and Aslam (2006).

Running Example Figure 2.2 provides the running example which is used throughout this thesis and is a representative for a standard query in the TRECVid setting. The depicted collection consists of recent broadcast news videos. Answers to the shown information need require video shots containing the U.S. President Barack Obama. The query is specified by two images and two query terms. The choice of the example images shows the difficulty of a user formulating a query in this way. Example image $q.s_2$, especially only shows the president as a detail of the image and is visually focused on the desert and mountain chain in the background. It is realistic that this example could have been the only picture a user could find for the formulation of his need, which would have led to poor search performance. On the other hand, example image $q.s_1$ is probably more suitable since it clearly shows the concept *US-Flag* which is useful to search for “President Obama”. Furthermore, the query text is supposedly easier to interpret and to formulate by the user.

2.3 Background: Multimedia Content Analysis

2.3.1 Video Segmentation

There are three reasons why videos are segmented prior to retrieval. First, a video can be multiple hours long and a user with a specific information need does only want to see the relevant parts. Second, it is necessary to make the content-based analysis computationally tractable. Finally, from the perspective of a retrieval model it is easier to operate on features of discrete units rather than continuous features over time.

The unit of a video shot, which is currently used by most video retrieval engines is defined by Hanjalic (2002) as follows: "A video shot is defined as a series of interrelated consecutive frames taken contiguously by a single camera and representing a continuous action in time and space". Smeaton et al. (2009) find in a large scale study of methods used by TRECVid participants, that current shot segmentation algorithms show sufficient performance to be employed in production systems.

Common shot lengths are around ten seconds and therefore they are suitable to fulfill the system oriented requirements of reducing computational complexity and to make the time dimension discrete. On the other hand, from a user perspective a shot is only a suitable result unit, if the information need is specific to a short time frame. However, longer, more semantic retrieval units are difficult to detect. Hsu et al. (2006) segment broadcast news shows into news items such that each shot belongs to exactly one news item. The underlying technique is a machine learning model which reidentifies the anchorman or woman of a news item based on training data, this segmentation method will be used in Chapter 6 for the retrieval of longer video segments. While this approach works well in the broadcast news domain, the semantic segmentation of arbitrary videos is still an unsolved problem.

Until now, the term *document* was used as an abstract retrieval unit. However, most concept-based video retrieval models exclusively use shots. Therefore, we will refer to a shot, if a statement is only true for this retrieval unit.

2.3.2 Low Level Feature Extraction

Low level feature extraction is a central system component in any multimedia retrieval engine. There are various kinds of features and each expresses a different aspect of a multimedia document. This section gives a brief overview of existing feature classes which are currently used for concept detection:

Visual features are the most commonly used features in concept detection Snoek and Worring (2009). For video shots, the visual features are normally extracted from few, but mostly one, key-frame(s) to reduce computational complexity. The features differ in two aspects:

- Low level features differ in *the part of the image* they describe. The options are: first, features which describe the whole image (which are called global features). Second, features which describe only a region or key point (which are called local features). For the local features there are two ways of selecting the regions or points to describe: while *dense sampling* uses all regions or key points of the image *Key point extraction techniques* try to select only interesting points. A popular detector for such key points is the Harris-Laplace detector (Harris and Stephens, 1988).

- Low level features differ in *what* they describe. Among the well-known descriptors are: color descriptors, texture descriptors and edge and shape descriptors (Bovik et al., 1990). Furthermore, more advanced descriptors are for example the scale invariant feature transform (SIFT) feature (descriptor) (Lowe, 2004; van de Sande et al., 2010).

Due to their better description of the image, local features with key point extraction are currently gaining popularity. Here, the number of descriptors among images can vary. However, machine learning algorithms, which are used for concept-based detection, operate on fixed vector lengths. Therefore, Sivic and Zisserman (2003) propose the bag-of-visual-words approach, which creates a fixed-size visual vocabulary, where each word is represented by groups of commonly eight pixels. The relative frequency of such words over the whole vocabulary is then used as a feature vector.

Audio features were only recently employed for concept detection. Portelo et al. (2009) and Peng et al. (2009) are among the first to use audio features for concept detection. The result of the extraction is the low-level feature vector, which is described as follows:

Definition 2.4. Let $\vec{LF}$ be the low-level features and let $\vec{lf}(d)$ be the low-level feature vector of the document d resulting from a low-level feature extraction process described in this above.

2.3.3 Concept Vocabulary

Prior to the detection of concepts the concept vocabulary has to be defined. There are three main aspects for the definition of such a vocabulary. First, the concepts must be useful to answer the queries with the data contained in the collection. Second, the concepts in a vocabulary should also be detectable by a computer. For example, the concept *Catastrophe* is probably a good concept for a search in a news collection, however, it is not likely to be detectable. Finally, since most detector methods use examples of concept occurrence (positive example) and concept absence (negative examples) the selection of concepts also depends on the practical feasibility of providing such examples, because it requires human labor and therefore financial resources (Ayache and Quénot, 2008).

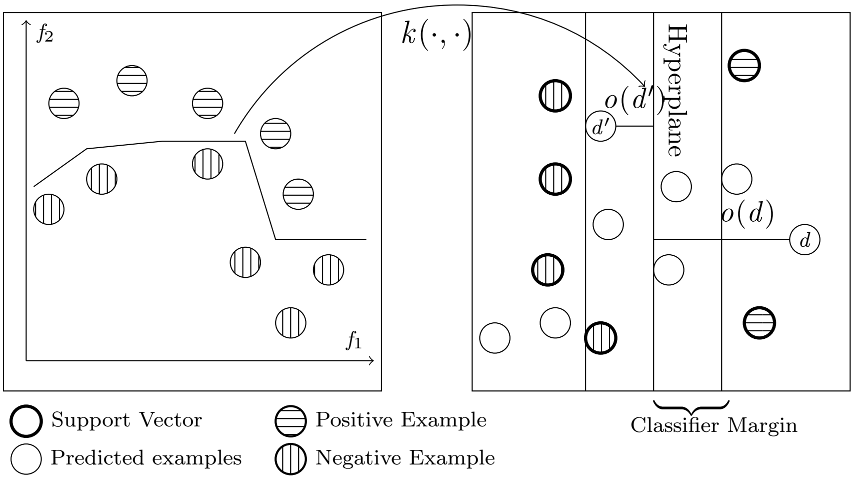


Figure 2.3: *SVM Classification: non-linear classification boundaries are projected with a kernel function $k(\cdot,\cdot)$ to a hyperspace where the hyperplane should divide positive and negative examples. The parameters of the hyperplane are set to maximize the classifier margin. The white balls on the right are predicted documents of which the concept occurrences are unknown. For such a document d, the only observable value is $o(d)$, the confidence score.*

Definition 2.5. Let $\mathcal{V}_C$ be the vocabulary of concept features C for which an information retrieval engine has concept detectors available.

2.3.4 Concept Detection

The task of a concept detector is to recognize the occurrence of a concept in a shot and is commonly performed using methods from machine learning. Virtually all detectors are trained on a development collection. The most important quality criteria of a detector is that it *generalizes* to other collections, see Yang and Hauptmann (2008a).

Currently, support vector machines (SVM) (Vapnik, 1999) are the most frequently used classifiers for the detection of concepts (Snoek and Worring, 2009; Yang and Hauptmann, 2008a). Therefore, this section focuses on the description of this concept detector method. For a more in-depth description of the state-of-the-art in concept detection the reader is referred to Snoek and Worring (2009).

An SVM operates on data points where each point is described by a feature vector. For concept detectors, the data points are shots and the feature vectors are the low-level features. An SVM operates in two phases: the training phase and the prediction phase. Figure 2.3 gives an example of the working of an SVM which is reduced to two dimensional feature vectors for display purposes. On the left, positive and negative training examples are shown. As the decision boundary which separates positive and negative examples is non-linear, the coordinates of the feature vectors are projected into a so-called hyperspace where the separation is easier. This projection is done via a kernel function, $k(\cdot,\cdot)$, which takes two feature vectors as arguments. The most commonly used kernel function for concept detection is the Gaussian radial basis function. The reader is referred, for example, to Bishop (2006) and Snoek and Worring (2009) for more information on the topic of kernel functions. One of the arguments of the kernel function is a so-called support vector. Support vectors are used to define the projection and are selected during the training phase. The objective for the selection of support vectors is the maximization of the classifier margin (which is also referred to the cost parameter), indicated in Figure 2.3. Since there are often more negative than positive examples, support vectors of the positive class can be assigned a higher weight to increase their influence. Therefore, a concept detector is fully specified through the training data, the kernel function with its parameters, the cost parameter and the weights of the support vectors.

The settings are normally found by iterating over different parameter values to select the parameter set which optimizes a certain performance measure. Often this measure is the rate of correct classifications. However, due to a low detector performance, concept detectors are usually trained to optimize the average precision. The measurement is determined through cross-validation (Bishop, 2006) to prevent overfitting to a certain part of the test data.

The right side of Figure 2.3 shows the prediction phase. Here, the shots whose concept occurrences should be predicted are projected into the hyperspace, using their feature vectors and the previously defined kernel function. The confidence score O of a shot with feature vector $\vec{lf}(d)$, is the distance between the shot coordinates in the hyperspace and the support vectors, see also Figure 2.3. It is calculated as follows (Vapnik, 1999):

$$o(\vec{lf}(d)) = \sum_{i}^{n} y_i \, \alpha_i \, k(\vec{lf}(d), \vec{lf}(sv_i)) \; + b \tag{2.1}$$

Here, n is the number of support vectors, $y_i \in \{-1, 1\}$ is the label (concept occurrence or absence)[2] and α_i is the weight for the ith support vector with low-level features $\vec{lf}(sv_i)$. Furthermore, b is a constant which defines an offset to the hyperplane. For simplicity, we drop the dependency of O of the low-level feature vector $\vec{LF}$ and assume that it is directly dependent on the document d in the rest of this thesis.

If the SVM is treated as a binary classifier, a decision criterion is used to derive a classification decision (occurrence or absence) from the confidence score O. However, in concept-based retrieval the confidence score is normally used directly since the classification errors are commonly too numerous to provide sufficient retrieval performance.

[2] The label values -1 and 1 are commonly used in machine learning and have the advantage that the negative class can also have negative influence in discriminative models. However, where this is not needed this thesis uses the labels 0 and 1 to conform with text retrieval notation.

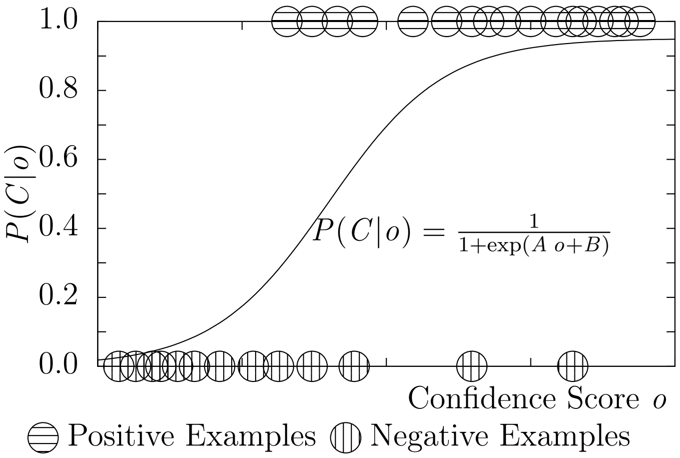


Figure 2.4: *The positive and negative examples are training data. Assuming a good SVM model the positive examples will be denser distributed in positive areas of the confidence scores o. The posterior probability follows a sigmoid function. The Figure is similar to Platt (2000).*

Definition 2.6. Let O_C be the confidence score feature of a concept detector for concept C whoes calculation is defined in Equation 2.1. Furthermore, let $\mathcal{V}_O$ be the vocabulary of all confidence score features available to the retrieval engine.

From Confidence Scores to Probabilities The confidence score $o(d)$ of a document d from Equation 2.1 depends on the trained detector model and can take different ranges among concepts in reality. Many retrieval functions require comparable, normalized scores. The most common normalization, which is also used in the retrieval models in this thesis, is the use of a probabilistic measure for the class membership of a shot. Platt (2000) proposes that the posterior probability of the concept occurrence C follows a sigmoid function of the confidence score $o(d)$ of shot d. This proposition is widely adopted among researchers. The discriminative model of Platt's posterior probability has following definition:

$$P_\Omega(C|o) = \frac{1}{1 + \exp\left(A\,o + B\right)} \tag{2.2}$$

Here, Ω is the probabilistic event space of the posterior probability function which is further defined below. After the SVM training phase, the parameters A and B of the sigmoid function are fitted to the confidence scores of the training collection.

Figure 2.4 shows a visualization of Platt's fitting method to train the parameters A and B of the sigmoid function. The x-axis shows the confidence scores and the y-axis the posterior probability. At the top the positive examples of the training set are shown. The negative examples are at the bottom. For a reasonable detector the densities of positive examples are higher at bigger confidence scores and the density of negative examples will be higher at lower confidence scores.

In the original work by Platt (2000), the probabilistic event space, on which the probability measure is defined, is not explicitly described. Because it is of importance for later argumentation, the assumptions made in this thesis are defined.

Definition 2.7. Following (Bishop, 2006, Sec. 1.2.1) it is assumed that the confidence score O is continuous and defined on the probability event space Ω, which is assumed to contain an infinite number of documents, being the universe of documents.

Multiple Concepts Often, the probability of the occurrence or absence of multiple concepts is considered. Although multi-class SVMs exist (Duan and Keerthi, 2005), which jointly model all occurrence and absence combinations, in current concept detector the posterior probabilities of two concepts C_1 and C_2 are modeled independently (Snoek and Worring, 2009). The joint posterior probability for a document with confidence scores o_1 and o_2 is calculated as follows.

$$P_\Omega(C_1, C_2|o_1, o_2) = P_\Omega(C_1|o_1)P_\Omega(C_2|o_2) \tag{2.3}$$

The following independence assumptions between the random variables C_1, C_2, O_1, O_2 are made in the above equation.

$$\begin{align} P_\Omega(C_1|O_1, O_2) &= P_\Omega(C_1|O_1) \tag{2.4}\\ P_\Omega(C_1|C_2, O_1) &= P_\Omega(C_1|O_1) \tag{2.5}\end{align}$$

Here, C_1 and C_2 are interchangeable and Equation 2.3 can be derived as follows.

$$\begin{aligned} P_\Omega(C_1, C_2|O_1, O_2) &= P_\Omega(C_1|C_2, O_1, O_2)P_\Omega(C_2|O_1, O_2)\\ &\underbrace{=}_{Eq.2.4} P_\Omega(C_1|C_2, O_1)P_\Omega(C_2|O_2)\\ &\underbrace{=}_{Eq.2.5} P_\Omega(C_1|O_1)P_\Omega(C_2|O_2)\end{aligned}$$

The first independence assumption, see Equation 2.4, states that the confidence score O_2 for concept C_2 does not influence the confidence of the detector for concept C_1 given the confidence score o_2. This assumption could be violated by two concepts *Street* (C_1) and *Car* (C_2), if the occurrence of *Cars* is dependent on the confidence scores of *Streets*, for example if *Cars* occur more often for certain confidence scores of *Street.* The second independence assumption, see Equation 2.5, can be thought of as semantic independence.

For example, the semantic independence could be violated for two concepts that have an 'is-a' relationship. For example, for the concepts *Mr. Obama* (C_1) and *Person* (C_2), because $P_\Omega(C_1|\bar{C}_2, o_1)$ will always be zero, no matter what value o_1 takes. The reason is that in no shot will concept C_1 ever occur while C_2 is absent.

Current research in concept detection also tries to explicitly model such dependence, see for example (Snoek and Worring, 2009, Sec. 2.5) and (Wei et al., 2009). However, the independence assumptions from Equation 2.4 and Equation 2.5 also have practical advantages: the complexity of such models is smaller because they depend on fewer parameters. Moreover, to create a detector model for multiple concepts the number of required training examples increases rapidly – especially since many combinations will be only rare and many training examples have to be inspected until a sufficient number of each combination is found. This phenomenon is well-known and commonly termed "The curse of dimensionality", see for example Bishop (2006).

The Urn Metaphor Following the practice of describing probabilistic processes with simplified examples, as done by Robertson (2005) using Stars and Planets, this thesis uses a more down-to-earth metaphor of balls residing in urns. This metaphor will be used and extended in the course of this thesis.

Thought Experiment 2.8. The Urn Metaphor: *Since concept detectors have the purpose of predicting the concept occurrence in any future document of the universe of documents Ω, we imagine an infinite number of balls where each ball represents a document. If a single concept is considered, each ball d has a known confidence score $o(d)$ for concept C. Furthermore, each ball has a color which indicates the occurrence or absence of the concept, for example red (for concept occurrence) and blue (for concept absence). However, the colors of the balls are unknown, let us assume they are wrapped in intransparent plastic. Now, for any confidence score value o, the posterior probability of a concept occurring given this confidence score, $P_\Omega(C|o)$, can be imagined as follows: if all balls with the confidence score o are put into an urn, labeled by the confidence score o, then $P_\Omega(C|o)$ is the proportion of red balls in this urn.*

If multiple concepts are considered, the balls carry multiple confidence scores and colors. For an example with two concepts, the colors red and green stand for the occurrence of the two concepts C_1 and C_2 respectively and blue and yellow for the absence of C_1 and C_2. Balls with their two dimensional confidence score vector $\vec{o} = (o_1, o_2)$ are then put into the urn labeled with $\vec{o}$. Now, for any two dimensional confidence score vector $\vec{o}$, the posterior probability of a color combination of multiple concepts $P_\Omega(C_1, C_2|\vec{o})$ can be imagined as the proportion of this color combination in the urn $\vec{o}$.

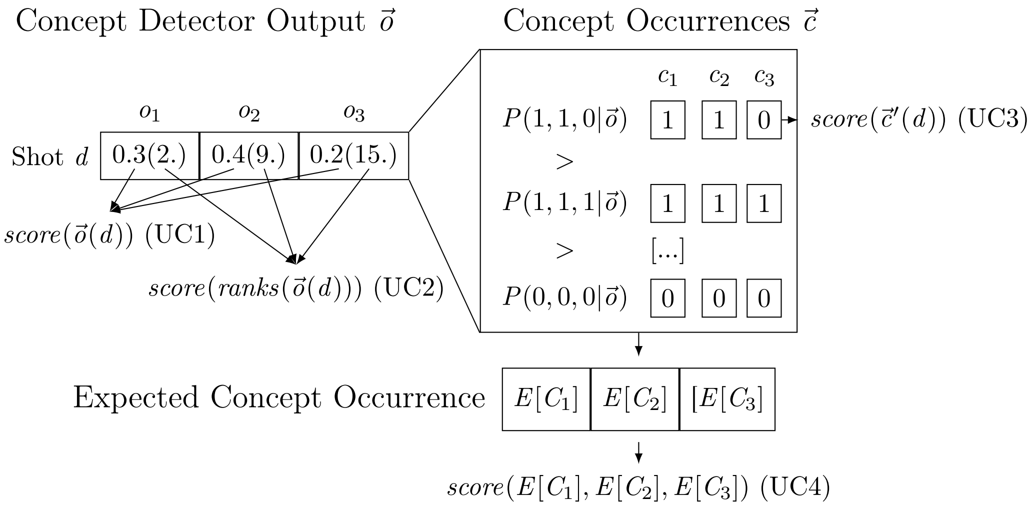

Concept Detector Output $\vec{o}$
Concept Occurrences $\vec{c}$
o_1 o_2 o_3
Shot d 0.3(2.) 0.4(9.) 0.2(15.)
$score(\vec{o}(d))$ (UC1)
$score(ranks(\vec{o}(d)))$ (UC2)
c_1 c_2 c_3
$P(1,1,0|\vec{o})$ 1 1 0
$score(\vec{c}'(d))$ (UC3)
>
$P(1,1,1|\vec{o})$ 1 1 1
> [...]
$P(0,0,0|\vec{o})$ 0 0 0
Expected Concept Occurrence $E[C_1]$ $E[C_2]$ $[E[C_3]$
$score(E[C_1],E[C_2],E[C_3])$ (UC4)

Figure 2.5: *Classes of uncertainty treatment (UC) in concept-based retrieval functions: confidence score value base (UC1, Section 2.4.2), Confidence score rank based (UC2, Section 2.4.3), Best-1(UC3, Section 2.4.4), Expected Concept Occurrence (UC4, Section 2.4.5).*

2.4 Concept Retrieval Functions

This section reviews existing retrieval functions of automatic concept-based retrieval engines. Instead of enumerating a description of retrieval functions, they are discussed in regards to the problem P1, *Document Representation Uncertainty*, see Section 1.3.

2.4.1 Desirable Properties and Classification

In order to guide the discussion, the following desirable properties of retrieval functions are proposed, which help to solve the problem $P1$.

- **Reasonable interpretation under perfect detection**: it is desirable, that the formula of a retrieval function has a reasonable interpretation in the (theoretical) case that the concept detectors were certain. In this case we assume an output of $o(d) = 1$ if the concept occurs and $o = 0$ if does not.

- **Modeling of concept absence in relevant documents**: an information need will often require multiple concepts for its evaluation. However, not all used concepts will always occur in relevant documents. Therefore, it is desirable, that a retrieval function should model the case where a document is relevant even though a concept does not occur.

Furthermore, a classification scheme of concept-based retrieval functions is proposed, which is based on how the functions treat the uncertainty of the concept detectors. The example scenario in Figure 2.5 shows a shot d and an example of the two possible document representations: on the left, the confidence scores $\vec{O} = (O_1, O_2, O_3)$ of shot d are shown. The rank of the confidence scores within the collection is shown in brackets. On the right, the possible concept representations $\vec{C} = (C_1, C_2, C_3)$ of the document d are shown, ordered by their posterior probability of the representation after observing all confidence scores, $P_\Omega(\vec{C}|\vec{o})$. At the bottom, a new document representation of expected concept occurrences is shown, which is described in Section 2.4.5.

This thesis identifies four different classes of how concept-based retrieval functions can handle the uncertainty of confidence occurrences (UC1-4). The classes are shown in connection with the used representation. In the following sections, important retrieval functions from each class are described.

2.4.2 Confidence Score Value Based (UC1)

Most current concept-based retrieval functions explain the relevance of a document based on a document representation of confidence scores (Snoek and Worring, 2009).

The retrieval function CombSUM, originally proposed by Fox and Shaw (1993) for meta-search, is defined as follows.

$$retfunc_{CombSUM}\langle\vec{O}, w\rangle(\vec{o} : dom(\vec{O})) = \sum_{i=1}^{n} w_i o_i \tag{2.6}$$

Here, w_i is the weight of the confidence scores of concept i. In the unweighted case ($w_i = 1$, $\forall i$) and perfect detection, *CombSUM* results in a count of the occurring concepts and therefore important concepts have the same influence as less important concepts.

Zheng et al. (2006) propose the pointwise mutual information weight (PMIWS) which is a weighted instance of CombSUM. The retrieval function is defined as follows.

$$retfunc_{PMIWS}\langle\vec{O}, w\rangle(\vec{o} : dom(\vec{O})) = \alpha \sum_{i}^{n} \log\left(\frac{P(C_i|R)}{P(C_i)}\right) P_\Omega(C_i|o_i) \tag{2.7}$$

Here, $w_i = P(C_i|R)$ is a query-specific weight and $P(C_i)$ is the concept prior which will both be explained in Section 2.5 and α is normalization constant. The combination of the two parameters by $log(P(C_i|R)/P(C_i))$ is the pointwise mutual information weight between concept occurrence and relevance. Under perfect detection, the retrieval score value of PMIWS is the sum of the pointwise mutual information of the occurring concepts. The case that a concept could be absent in relevant shots is not modeled. Therefore, under perfect detection the score for documents where all concepts occur is reasonable; since they should be ranked highest. However, the mutual information for a document representation where some concepts do not appear is not modeled while they will often be non-zero. Therefore, the interpretation for representations with concept absences is not clear.

The CombMNZ method is defined as follows.

$$\mathit{retfunc}_{\mathit{CombMNZ}}\langle\vec{O}\rangle(\vec{o} : \mathit{dom}(\vec{O})) = \prod_{i=1, o_i>0}^{n} o_i \tag{2.8}$$

Under perfect detection, CombMNZ converges to a logical OR in Boolean search (Chowdhury, 1998) which is accepted in the text retrieval community to be an undesirable retrieval function (Hiemstra, 2001).

Yan (2006) proposes the Probabilistic Model for combining diverse Knowledge Sources in Multimedia (PKSrc). The proposed retrieval function is a discriminative model of the posterior probability of relevance given the observation of the confidence scores. It is proposed that the posterior probability follows the shape of a sigmoid function and is defined as follows.

$$\mathit{retfunc}_{\mathit{PKSrc}}\langle\vec{O}, w\rangle(\vec{o} : \mathit{dom}(\vec{O})) = P_{\Omega}(R|\vec{o}) = \frac{1}{1 + \mathit{exp}(-\sum_i^n w_i o_i)} \tag{2.9}$$

Here, w_i is the weight for the confidence scores of concept i. The weights are learned on the relevance judgments of similar queries and confidence scores in a training collection which will be explained in Section 2.5 where also more discussion of this model is provided.

2.4.3 Confidence Score Rank Based (UC2)

Retrieval models from the uncertainty class UC2 use the *rank* of the confidence scores in the collection as arguments to their retrieval function. The function is often derived from the Borda Count method, which originates from election theory in politics and was applied to concept-based video retrieval by Donald and Smeaton (2005). For example, the retrieval function is defined as follows.

$$\mathit{retfunc}_{\mathit{Borda-Count}}\langle\vec{O}, w\rangle(\vec{o} : \mathit{dom}(\vec{O})) = \sum_{i=1}^{n} w_i \; \mathit{rank}(o_i) \tag{2.10}$$

Here, $\mathit{rank}(o_i)$ is the rank of the confidence score in the collection $\mathcal{D}$. The fact that concepts could be absent in relevant shots is not modeled in this retrieval function.

Snoek et al. (2007) propose a variation of the Borda Count method in which they use one authoritative source (i.e. a concept ranking) and multiple secondary sources. Documents, which are not found in the ranking of the authoritative source before a cut-off value, are not considered further. For the rest of the documents the traditional Borda Count method is performed with weights consisting of the match of the concept to the query, the detector performance and prior of the concept. Therefore, the non-authoritative concepts have an implicit handling of the negative class, they can only be beneficial and are neutral upon absence.

2.4.4 Best-1 Representation (UC3)

Retrieval functions in the uncertainty class UC3 consider for each document only the most probable concept representation. Although there is no existing concept-based retrieval function of this class, the class is included because of the popularity of its counterpart in spoken document retrieval where the most probable spoken text is used as a document representation and achieves good results under good detection performance (Voorhees and Harman, 2000). A possible reason why uncertainty class was never used in concept-based retrieval is that concept detectors still have a low performance and important concept occurrences are easily missed in the most probable representation. This causes the retrieval performance to deteriorate. This outcome was also reported in Mamou et al. (2006) for spoken document retrieval with poor recognition performance.

For later comparison between methods from the different uncertainty classes the well-known binary independence model (BIM) (Robertson et al., 1981) is introduced. Because of the similarity of concept occurrences and index terms, see Section 2.2.2, the retrieval function is adapted to use estimated concept occurrences as arguments. Let the estimated concept occurrence C' be defined as follows.

$$c'(d) = \begin{cases} 1 & \text{if } P_\Omega(C|o) \geq 0.5; \\ 0 & \text{else.} \end{cases}$$

The adapted binary independence model is then defined as follows.

$$retfunc_{BIM}\langle \vec{C}', w\rangle(\vec{c}' : dom(\vec{C}')) = \sum_{i}^{n} w_i c'_i \tag{2.11}$$

with

$$w_i = \log \frac{p_i(1-\bar{p}_i)}{\bar{p}_i(1-p_i)} \quad \text{where} \quad p = P(C|R) \quad \text{and} \quad \bar{p} = P(C|\bar{R})$$

2.4.5 Expected Concept Occurrence (UC4)

Recently, document representations of expected term frequencies were successfully used in spoken document retrieval, see Chelba and Acero (2005) and Chia et al. (2008). Although there is no retrieval function in concept-based retrieval which uses the equivalent representation of expected concept occurrences explicitly, this thesis proposes that the language model retrieval function from Li et al. (2007) should be interpreted in this way. To see why, the derivation of the original model is reconstructed: the stated aim of the work by Li et al. (2007) is to resemble the Cross Entropy language model (RL) retrieval function proposed by Lavrenko (2004) which is *conceptually* defined as the Cross Entropy between a relevance and a document language model.

$$retfunc_{RL-THEORY}\langle\rangle(P(t_1|d),...,P(t_n|d)) = \sum_i P(t_i|R)log(P(t_i|d)) \quad (2.12)$$

Here, $P(t_i|R)$ is the term specific weight of the *relevance model* and $P(t_i|d)$ is the probability that a term is produced by the multinomial term distribution of the document d – which has to be interpreted as a theoretical document representation, since the document representation of draw probabilities $P(t_i|d)$ is unknown. Therefore, Li et al. (2007) propose the Jelinek-Mercer smoothing approach to estimate the draw probabilities (Hiemstra, 2001). In text information retrieval, $P(t_i|d)$ is estimated through the term frequencies $\vec{tf}_i$) and the document length dl of the document d and the collection $\mathcal{D}$. Therefore, the de facto document representations of the language model retrieval function are the term frequencies of the document and the operational retrieval function RL is defined as follows.

$$retfunc_{RL}\langle\vec{TF}\rangle(\vec{tf} : dom(TF), dl : \mathbb{N}) = \sum_i^n P(t_i|R)log\Big(\lambda\frac{tf_i}{dl} + (1-\lambda)P(t_i|\mathcal{D})\Big) \quad (2.13)$$

Here, $P(t_i|\mathcal{D})$ is the prior of obtaining term t_i in the collection. Now, when transferring this ranking function to concept-based retrieval, Li et al. (2007) make the assumption that $P(C|d)$ can be used equivalent to $P(t_i|d)$ from Equation 2.12. This results in the following ranking function.

$$retfunc_{LI}\langle\rangle(P(C_1|d),...,P(C_n|d)) = \sum_i^n P(C_i|R)log\Big(\lambda\ P_\Omega(C_i|o_i) + (1-\lambda)\ P(C_i|\mathcal{D})\Big) \quad (2.14)$$

Here, the following observation can be made.

(1) The ranking function prototype does not depend on the confidence scores $\vec{O}$, which are suddenly inserted in Equation 2.14, by assuming that $P(C|d) = P_\Omega(C|o)$.

(2) Equation 2.14 is a syntactic mixture of the theoretical retrieval function (Equation 2.12) and the operational retrieval function (Equation 2.13).

(3) There is a discrepancy between the two probabilities $P(C|d)$ and $P_\Omega(C|o)$: from the thought experiment 2.8 it can be seen that the probability $P(C|d)$ can only take two values: zero if the color beneath the intransparent plastic of the ball d is blue and therefore it is impossible to draw the color (concept) red ($P(C_i|d) = 0$), and one if the color beneath the

intransparent plastic was indeed red and therefore the document certainly produces red (the concept) ($P(C_i|d) = 1$). However, the probability $P_\Omega(C|o)$ can take any value between 0 and 1, since it states the proportion of the balls with the color red in the *urn* $\vec{o}$

The above observations lead to the conclusion that the definition of the retrieval function could be improved.

This thesis re-interprets the retrieval function as follows: in a shot, a concept can either occur or not and a shot is always of length one. Therefore, it is sufficient to consider the concept occurrence C to replace the fraction of term frequency and document length tf/dl in Equation 2.13. All we know about the document (ball) d is its urn o (confidence score) and the color beneath the intransparent plastic is unknown. Furthermore, probability theory states that the expected value of a random variable is a good estimator for its value. Therefore the expected concept occurrence is used instead of the true value of the concept occurrence which is either zero or one but unknown. The conditional expectation of a concept occurring given its confidence score is defined as follows.

$$E[C|o] = \sum_{c \in \{0,1\}} c \; P_\Omega(C = c|o) = P_\Omega(C|o)$$

This estimate is then plugged into the language model retrieval function from Equation 2.13 to arrive at the following.

$$retfunc_{ELM}\langle\vec{O}\rangle(\vec{o} : dom(\vec{O})) = \sum_{i}^{n} (C_i|R) log\Big(\lambda \; E[C_i|o_i] + (1-\lambda) \; P(C_i|\mathcal{D})\Big) \quad (2.15)$$

This interpretation has the following advantages over the one by Li et al. (2007): it is clearer what the arguments for the retrieval function are and that $E[C|o]$ is only an estimate for the possible values of C.

Under perfect detection, the expected value converges to the correct value of C and is in this case equivalent to the operational language model above, see Equation 2.13, which has proven to give good performance in text information retrieval. Furthermore, in the retrieval function ELM, the term $(1-\lambda) \; P(C_i)$ can be seen as a model for the case that a concept does not occur in relevant shots (Hiemstra, 2001).

However, there is also a disadvantage to this approach: although, it is intuitive that the expected value (a real number), is somehow a good estimate for the concept occurrence; the interpretation, why the use of this estimate in the language retrieval function, which is actually defined on integer frequencies, is unclear.

2.5 Concept Selection and Weighting

This section analyzes existing concept selection and weighting methods by the guidance of desirable properties which are derived from the problem P2, *Query Formulation Support*, see Section 1.3.

2.5.1 Desirable Properties

In order to solve the problem P2, a concept selection and weighting method should fulfill the following desirable properties.

- **Non-Expert User**: a retrieval engine should not require that a user name important concepts in the query text nor explicitly assign weights to those concepts. The user should not have to invest a lot of time and effort in formulating his query.

- **Collection specific selection and weighting**: while it is desirable that the selection and weighting *method* should work for any collections and information need, the concept selection and weighting for a *particular* information need and collection $\mathcal{D}$ should be specific to the relevant shots in collection $\mathcal{D}$.

The motivation of the two properties is further elaborated: the non-expert user requirement originates from observations of the development of information retrieval. While retrieval engines were originally built for experts (librarians) who have a thorough methodology of indexing and searching in catalogues, their biggest success was to enable common people to use (web-) retrieval engines. Here, a user is only required to type a few words (with hardly any consideration) and often receives search results of good quality. Today, even librarians use internet retrieval engines to do their investigations. As a result, users of multimedia retrieval engines should only be asked of a comparable effort to specify their query.

In order to see why it is desirable for a selection and weighting methods to be collection-specific, an example is given: in a collection of documentaries about U.S. presidents the concept *US-Flag* will probably not be useful to answer the information need “President Obama” since all presidents will be shown with this flag. However, in a collection of current broadcast news videos a *US-Flag* is probably a useful concept, since *President Obama* often occurs in documents with *US-Flags* and therefore the *US-Flag* distinguishes many relevant shots from the vast majority of shots.

The existing concept selection and weighting methods can be categorized into three classes. First, methods which only depend on the query. Second, collection based methods which analyze the data of the search collection to identify useful concepts. Finally, query class based methods which assume that a query belongs to one particular class of queries and the selection and weighting is inherited from this class.

2.5.2 Query Based Methods

Methods from this class focus on the query to derive the concept selection and weightings.

Example Image Based

The selection and weighting methods discussed in this section operate exclusively on provided example images. Therefore, the desirable property, that a user can specify their query by typing words, is not met.

For the PMIWS method (Equation 2.7), Zheng et al. (2006) consider a single example image $q.s$ as the query. The probability $P_\Omega(C|o(q.s))$ is used as the probability that a concept occurs in relevant shots $P(C|R)$. A selection is not done and all concepts are used for every information need. The assumption that $P(C|R) = P_\Omega(C|o(q.s))$ makes it difficult for a user to provide good query examples: for example, if the user presents an example image where "President Obama" is shown in the desert, a *Desert* detector could have a high probability. This, in turn, would cause the score function to rank shots with deserts higher which was not the intention of the user. Furthermore, the concept prior $P(C)$ introduces a collection-specific weighting component.

Li et al. (2007) propose to first sort concepts by a measure similar to the well-known tf-idf weighting scheme from text retrieval (Spärck-Jones, 1972) and then select the first n concepts from the list. The measure is defined for a concept C and example images $\{q.s_1, ..., q.s_n\}$ as follows.

$$\text{c-tf-idf}\,(C,q) = \underbrace{\mathit{freq}(C,q)}_{\mathit{tf}}\,\underbrace{\mathit{log}\Big(\frac{N}{\mathit{freq}(C)}\Big)}_{\mathit{idf}}$$

with

$$\mathit{freq}(C,q) = \frac{\sum_{i=1}^{n} P_\Omega(C|o(q.s_i))}{n} \qquad \text{and} \qquad \mathit{freq}(C) = \sum_{d\in\mathcal{D}} P_\Omega(C|o(d))$$

Here, $\mathit{freq}(C,q)$ is the estimated concept frequency in the query images and $\mathit{freq}(C)$ is the estimated concept frequency in the collection. Since multiple query images are used, the unwanted effect of randomly occurring concepts for the concept selection and weighting is less strong compared to PMIWS, see above. The language model weight, $P(C|R)$, from Equation 2.15 is calculated in the same way as $\mathit{freq}(C,q)$. The inverse document frequency *idf* component makes the selection and weighting method collection-specific. For a specific query, a concept selection is performed and the first n concepts from the sorted list of concepts by descending c-tf-idf(C,q) are selected.

Query Text Based

Methods from this class are mainly based on the query text provided.

Term Matching Based Methods Chang et al. (2006) propose a method which selects the concepts which are directly named in the query. Therefore, this method assumes that the user knows about the concept vocabulary, which is not user friendly. Furthermore, this does not include a component representing concept importance.

Rule Based Methods Natsev et al. (2007) propose a rule based method which statically maps query terms to concepts using rules, implemented into the retrieval engine. For example, if a retrieval engine receives a query containing the term 'president', and a rule states 'President'→*Person*; the concept *Person* is selected to be used for the information need. However, instead of concept names, the user is now required to know the terms leading to the concept selection which is only a slight improvement over the term matching based method. The method does not propose a weighting scheme.

WordNet Based Methods In many concept selection methods, the WordNet thesaurus (Fellbaum, 1998) is used to find useful concepts, see for example Hauff et al. (2007); Haubold et al. (2006); Snoek et al. (2007). Here, concepts are connected to the WordNet graph. For a specific query, the query terms are located in the thesaurus and a measure of the semantic distance between the query and each concept is calculated. In the literature, multiple semantic distance functions were proposed, see Hauff et al. (2007) for an overview. For example, Wu and Palmer (1994) define the semantic distance between two concepts C_i and C_j as follows.

$$WUP(C_i, C_j) = \frac{2D(p_{ij})}{L(C_i, C_j) + 2D(p_{ij})}$$

Here, p_{ij} is the common ancestor of the two concepts, $D(\cdot)$ is the depth in the WordNet hierarchy and $L(\cdot)$ is the length of the path between the two concepts. Afterwards, the semantically closest concepts to the query nodes are selected. However, the distance reflects the linguistic relation and not the relationship of the concepts' occurrences in relevant shots. Therefore, this weight is not collection-specific. For example, the concept *George Bush* could be close to the query node for the term 'President', but it is unlikely that it will help to find relevant shots to the information need "President Obama". Furthermore, the semantic distance measure is not always a suitable weight for many retrieval functions, see for example our previous work Aly et al. (2008a).

Text Representation Methods In this selection and weighting class, the query text is used with a text information retrieval engine storing textual concept descriptions to calculate a retrieval score which is then used for concept selection and weighting.

Snoek et al. (2007) use the query document similarity of the vector space model from Salton et al. (1975) to rank concept documents for a textual query, where a concept document is a short textual description of a concept. Here, the concept weights are the retrieval scores values from the vector-space model. However, the shortness of the descriptions limits the user in their query formulation. Furthermore, the vector space similarity score of a concept document for an information need is problematic to use because its interpretation does not match with the ones in known retrieval functions.

Similarly, Hauff et al. (2007) also use text retrieval, although with a collection of longer concept descriptions, where Wikipedia articles and concatenated WordNet Glosses are investigated. For an information need, a text search is performed using the query text. The returned text retrieval scores of the concept descriptions are used to measure the importance of the concept described by the text. The elaborate and up-to-date character of Wikipedia articles allows the user to formulate its query more freely. For example, if a user typed in only 'Obama' chances are that in the Wikipedia articles of the concept *President* the term 'Obama' occurs and therefore this useful concept can be found which probably would not have been found by most other methods. On the other hand, similar to the vector space similarity, the score provided by the text retrieval engine is difficult to interpret as a weight in concept-based retrieval models.

2.5.3 Collection Based Methods

Collection based methods derive selection mechanisms based on the content of the search collection.

Statistical Corpus Analysis Natsev et al. (2007) propose the statistical corpus analysis concept selection method which selects useful concepts through co-occurrences of spoken terms and concepts in the search collection. For this, the confidence scores in the search collection are transformed to binary classifications and the automatic speech recognition output is assumed to be perfect. Then, a likelihood ratio test called $\mathcal{G}^2$, see Dunning (1993), is used to decide whether a significant correlation between a term and a concept exists. For a specific information need, concepts which are strongly correlated with the query terms are selected and a normalized version of the correlation coefficient is used as a concept weight. This strategy is collection dependent, since only concepts which co-occur in the collection with query terms are selected. However, there are also disadvantages: first, the use of binary classification and the assumption of a perfect automatic speech recognition output are only reasonable with good detector performance, which is often not the case in current systems. Second, the language usage and therefore the query terms used by the speakers in the video and by the user can be different. Finally, information needs – also those expressed by text – can concern requests for visual information, for example *President Obama waving*, which will not necessarily be reflected in the speech transcript and therefore useful concepts will not be found.

Semantic + Observability Wei et al. (2008) propose a combination of a WordNet based method (the semantic space) and a collection based method (the observability space) for concept selection and weighting. The *semantic space* is a vector space (not to be confused with the vector space from Salton et al. (1975)) with a WordNet distance measure as its components. The *observability space* contains the correlations between the confidence scores of all concepts in the collection. For example, the concepts *President* and *Person* are related in the semantic space and the concept *Car* and *Road* are correlated in the observability space because their confidence scores are correlated in the collection. For a specific information need, for each query term the concept with the smallest WordNet distance is selected. Additionally, these concepts are expanded with the highly correlated concepts in the observability space.

However, besides the problem inherited from the uses of WordNet distances, see Section 2.5.2, another problem of this method is that a user might especially ask for things which are normally not observable together. For example, if the concept *Mr. Obama* appears to 99% with the *U.S. Flag* this method searches for both concepts even though the information need was *President Obama in the desert.*

2.5.4 Query Classes

The usage of query classes is a widely used selection and weighting method in content-based multimedia retrieval (Yan et al., 2004; Huurnink, 2005). A query class determines the concept selection and weights for a set of queries. A retrieval engine has a limited set of query classes and each query is assumed to belong to exactly one class. For a query, the previously defined selections and weights are used for the ranking. The methods, which are derived from this general idea, differ in three aspects. First, how the query classes are defined. Second, how the concept selection and weighting in each class is performed. Finally, how a class is assigned to a new information need.

Static Query Classes Yan et al. (2004) propose the use of a static choice of query classes per class. Here, the set of query classes are defined by humans after investigating existing queries. The selected concepts and their weights per class are then optimized, based on the confidence scores in the training collection and provided relevance judgments. A new query is assigned to a query class either manually or automatically through the employment of natural language processing and named entity extraction.

Performance and Semantic based Query Classes Kennedy et al. (2005) propose a different query class scheme designed using the weighted CombSUM ranking function. A set of relevance judgments for existing training queries are used to cluster queries by their performance and linguistic features. Afterwards, each cluster is used as a query class and the optimal concept selections and weightings for each query class are determined by a full grid search, in which a large number of combinations of concept selections and weights are tried. For a new information need, a classifier based on a learned distance of the linguistic features of this query compared to the training queries is used to determine the query class of this need.

Probabilistic Latent Query Analysis The probabilistic Latent Query Analysis (pLQA) framework, proposed by Yan and Hauptmann (2006), determines the weights for the PKSrc ranking function in Equation 2.9. Concepts are selected and the query class dependent weights $\vec{w}$ for all query classes are determined. However, instead of statically assigning the query to a query class, the query class is assumed to be latent. The probability distribution, to which class a query belongs is then learned on the basis of training queries using a set of query features $\vec{qf}$. Example features are the binary features "A person is named" or "About sport" which are derived from the query text through natural language processing. Yan and Hauptmann (2006) propose two methods to learn the probability distribution over the query classes z given query features $\vec{qf}(q)$, $P(z|\vec{qf}(q))$. First, the Adaptive probabilistic Latent Query Analysis (ApLQA), which uses a soft-max fusion for the estimation. Second, the Kernel probabilistic Latent Query Analysis (KpLQA), where kernel density estimation is used to estimate the probability distribution.

Regardless of the learning method for the query class membership distribution, the retrieval function is then calculated by marginalizing over the latent query class variable z.

$$\begin{aligned} &retfunc_{PKSrc,pLQA}\langle \vec{O}, \vec{QF}, w\rangle(\vec{o}, \vec{qf}) = \\ &\qquad P(R|\vec{o}, \vec{qf}) = \sum_{z} P(R|\vec{qf}, \vec{o}, z)P(z|\vec{qf}) \end{aligned} \tag{2.16}$$

Here, w is a function assigning each class a weight vector and $P(R|\vec{qf}, \vec{o}, z)$ is the probability described in Equation 2.9 which depends on the class dependent weights w_z.

Although it is claimed in Yan (2006) that the variable R (y in the original notation) has the same definition as in Robertson (1977) this is strictly speaking not the case. In the work of Robertson (1977) the variable R is defined separately for each information need. However, the learning process over multiple information needs hints that the method actually has to be seen as an instance of Model 0 from Robertson et al. (1982) in which the probability of relevance is calculated on similar information needs and similar documents (all those with the same document representations $\vec{qf}$ and $\vec{o}$). In the PKSrc,pLQA model, the probabilistic event space is the cross product of all documents and all information needs instead of a single information need q and all documents. Robertson et al. (1982) and Bodoff and Robertson (2004) find that instances of Model 0 need excessive amounts of training data. This makes it questionable whether enough training material can be acquired for the successful application of the PKSrc,pLQA model.

2.6 Summary and Discussion

This chapter reviewed existing state-of-the-art concept-based retrieval models. In this review the two components of a retrieval model, the retrieval function and a concept selection and weighting method were discussed separately.

Retrieval Functions In Section 2.4, the existing retrieval functions were discussed. The following desirable properties of any retrieval function were identified. First, under perfect detection of concepts, the interpretation of the retrieval function should be reasonable. Second, a retrieval function should model that selected concepts could be absent in some relevant shots.

The existing retrieval functions were categorized into four classes with regards to how they treat the concept detector uncertainty (UC). The findings per class will now be summarized and discussed.

Confidence score value based (UC1) This class comprises most existing retrieval functions. However, Snoek and Worring (2009) found that it is difficult to set query-specific weights for the retrieval functions of this class. Some retrieval functions did not have a clear interpretation under perfect detection since they amounted to counting concept occurrences or Boolean queries, which is known to yield poor performance. Finally, none of the retrieval functions of this class modeled the case that documents could be relevant despite the absence of a considered concept.

Confidence score rank based (UC2) The only retrieval function of this class, the Borda Count method, differs mainly from members of the uncertainty class (UC1) in that it considers the ranks of the confidence scores instead of their absolute values. This uncertainty class has similar drawbacks to the confidence score value based uncertainty class (UC1).

Best-1 Representation (UC3) For this class, this thesis introduced the binary independence model using a concept occurrence document representation due to the parallel between automatic speech recognition and concept detectors. It was proposed that this class was never used in today's concept-based retrieval because the performance of the detectors is still low and therefore the classification performance, which is needed by this method, is low. However, it was proposed that setting the weights of these methods is easier than with confidence score based retrieval functions, since the concept occurrences of important concepts in relevant shots are not

Expected concept occurrence (UC4) This uncertainty class was also not mentioned in the literature. However, because of the parallel to recent retrieval functions in spoken document retrieval, the work on language concept models by Li et al. (2007) was re-interpreted to consider expected occurrence of a concept in a single document in the retrieval function. Under perfect detection this ranking function is equivalent to a language model ranking function, inheriting its scientific motivation. However, the expected term occurrences were used as concept occurrences which, from a theoretical standpoint, can only be zero or one.

Concept Selection and Weighting Section 2.5 discussed existing concept selection and weighting algorithms. Desirable properties of such algorithms were identified as the following. First, non-expert users should be considered and should not be required to have much knowledge of the collection or spend much effort in the query formulation. Second, the selection and weighting should be done collection-specific.

The concept selection and weighting methods were categorized by the data they operated on.

Query based methods Methods from this class perform selection and weighting only under consideration of the query. Therefore, all methods in this class were collection independent and many of them required knowledge of the concept vocabulary or required a considerable user effort during the query formulation.

Collection based methods Methods from this class are based on statistics from the search collection and predict useful features. Therefore, methods from this class select concepts in a search collection-specific manner. However, current methods significantly limit the user in their query formulation process: for example, Natsev et al. (2007) used the co-occurrence of spoken words and concepts in the search collection to connect query words with good concepts. However, this requires good detection performance and since terms from the spoken text are used the supported information needs are limited to the ones which look for shots relevant to the ones where the query words were spoken. Another method from Wei et al. (2008) selects additional concepts if their confidence scores are correlated with concepts already selected in the search collection. However, this can lead to the effect that only popular correlations are used. For example, the confidence scores of the concept *Car* are probably correlated with the confidence scores of *Road*. Therefore, if for a query the concept *Car* was selected the concept *Road* will be automatically included, which will lead to poor search results if looking for the information need “Cars in a desert race”.

Query Class based Methods from this class assume that an information need always belongs to a single query class and concepts and weights are trained for each class. The weights for the query classes were most often trained on the development collection. However, even a large number of query classes will limit the user in his query formulation since his information need is put into a mainstream category (the query class). An exception is here the Probabilistic Latent Query Analysis method from Yan and Hauptmann (2006), which uses a probabilistic assignment of a query to its query classes. However, the resulting retrieval model was identified to be an instance of the Model 0 from Robertson et al. (1982), which was reported to need vast amounts of training data, which was probably why its research was abandoned.

Final Summary Most current state-of-the-art retrieval functions still have a limited treatment of uncertainty and that current concept selection and weighting methods either require too much effort from a user or are not collection-specific or both.

Chapter 3

Uncertain Representation Ranking Framework

3.1 Introduction

This chapter presents the Uncertain Representation Ranking (URR) framework. The framework reuses text retrieval functions with concept-based document representations. It caters for the detector uncertainty by including the probability distribution of concept occurrence combinations given their confidence scores. Chapters 5 and 6 are applications of the URR framework and show the merits of this approach. Because the two chapters use different retrieval functions and different features (concept occurrences and concept frequencies) the URR framework is defined on an abstract document representation, which will be made concrete in the chapters applying the URR framework.

This work is not the first to treat uncertainty in information retrieval. In probabilistic indexing, where it is assumed that librarians do not know the correct index terms for a document but the index terms are only probabilistically known, Croft (1981) was the first to use the expected score of the binary independence score function. However, Fuhr (1989) reported that the expected score of the binary independence score function was not rank equivalent to the probability of relevance ranking function and this line of thought was not continued. In the URR framework, the expected score of other score functions is a central component for ranking documents and this chapter explains why this is reasonable.

Furthermore, Wang (2009) proposes the Mean-Variance Analysis framework which considers uncertain scores in text retrieval and is based on the Nobel Prize winning Portfolio Selection Theory by Markowitz (1952), which optimizes the percentages of the available budget one should invest in particular shares. Wang (2009) transforms the problem of selecting percentages to the one of ranking documents. Similar to the Mean-Variance Analysis framework, the URR framework is also derived from the Portfolio Selection Theory and inherits its scientific rigor. However, there is also a difference between the URR framework and the previous two works:

- In the URR framework, a document with a known representation of concept occurrences has a correct score which is the result of a score function applied to the concept-based document representation. However, due to the poor concept detector performance, the document representation is uncertain and therefore the same holds for the correct score.
- In the Mean-Variance Analysis framework the document representation is known, but the correct score is not the result of the score function applied to the document representation, but is distributed around this value, for reasons which are not explicitly modeled. The same holds for the Portfolio Selection Theory, only that here the win of shares is considered instead of scores.

The remainder of this chapter is structured as follows. First, in Section 3.2 the parts of the Portfolio Selection Theory and the Mean-Variance Analysis framework which are used in the URR framework are described. Section 3.3 describes the proposed URR framework and shows the parallels to the work by Markowitz (1952) and Wang (2009). Finally, Section 3.4 ends this chapter with a summary and discussion.

3.2 Background: Uncertainty Treatment

This section introduces the original Portfolio Selection Theory by Markowitz (1952) and subsequently the Mean-Variance Analysis, which translates this theory to the problem of ranking documents under uncertain scores in text retrieval.

3.2.1 Portfolio Selection Theory

Markowitz (1952) proposes the Portfolio Selection Theory which provides a method to decide what percentages of the available budget to invest in which share. The term 'Selection' in the theory's name comes from the fact that the outcome of the decision can also result in zero percent of the budget for a particular share, which effectively de-selects the share from the portfolio. The aim of the theory is to maximize the overall win of the portfolio in the future. Therefore, the central formula of the theory is:

$$Win = \sum_{j=1}^{N} p_j \; Win(d_j) \tag{3.1}$$

with

$$Win(d_j) \geq 0 \text{ and } 0 \geq p_j \geq 1 \;\; \forall j \qquad \text{and } \sum_{j} p_j = 1$$

Here, Win is the random variable of the overall win of an investor in the future, which is the sum of the wins of the individual shares d_j, $Win(d_j)$, and the percentage p_j of the available budget invested in the share d_j. The theory now assumes that the following *components* (quantities) can be predicted by analysts:

(1) The *expected* win of share d, $E[Win(d)]$, is expressing "What win is to be expected from the share d?".

(2) The *variance* of the win of share d, $\text{var}[Win(d)]$, is expressing "How widely do the possible wins vary?".

(3) The *co-variance* between the win of two shares d_i and d_j, $\text{cov}[Win(d_i), Win(d_j)]$, expressing "How does the win of share d_i influence the win of share d_j?". For example, if company d_i is a computer manufacturer who buys an operating system from company d_j to preinstall it on its products, then the win of the two shares is probably positively correlated.

These components are also used in the Mean-Variance Analysis framework and the URR framework.

Efficient Portfolios As Markowitz (1952) notes, it is trivial to maximize the expected overall win from Equation 3.1 by only investing in the share with the highest expected win. On the other hand, Markowitz (1952) argues that this is not a reasonable strategy because of the associated risk associated with putting all money into a single share. Instead, the theory provides a geometrical procedure to obtain an optimal selection of investment percentage $(p_1, ..., p_N)^*$ for a mixture of the expected overall win, $E[Win]$ and its variance $\text{var}[Win]$:

$$E[Win] - b\ \text{var}[Win]$$

Here, b is the *risk parameter* representing the risk attitude of the analysts that steers the mixture. This gemoetric procedure was needed at the time the theory was published because of low computing capacities. However, today the underlying rationale can be expressed as an optimization problem. As the expectation of a sum is the sum of the expectations and the variance of a sum can be expressed as the variance and co-variances of its summands, the following expression formulates the optimization problem, see Wang (2009) for a derivation:

$$
\begin{aligned}
(p_1, ..., p_N)^* \;&=\; \underset{p_1,...,p_N}{\operatorname{argmax}} \sum_{j=1}^{N} p_j \; E[\,Win(d_j)] \\
&-\; b \,\Big[\Big[\sum_{j=1}^{N} p_j^2 \;\mathrm{var}[\,Win(d_j)]\Big] \\
&+\; \Big[\sum_{j=1}^{N} \sum_{k=1,k\neq j}^{N} p_j p_k \;\mathrm{cov}[\,Win(d_j),\, Win(d_k)]\Big]\Big] \qquad (3.2)
\end{aligned}
$$

with

$$0 \geq p_j \geq 1 \;\; \forall j \qquad \text{and} \; \sum_j p_j = 1$$

Here, $(p_1, ..., p_N)^*$ are the optimal percentages which should be invested in the corresponding shares. If $b > 0$, the analyst is *risk-averse* and therefore try to spread the percentages among many shares. As stated earlier, the parameter setting $b = 0$ is unreasonable because all available budget would be invested in one share. In the case of $b < 0$ the analysts like to take risks, which is informally called *risk-loving* here.

Example In order to provide an insight into the workings of the Portfolio Selection Theory, an example is given. Let us assume two stock markets: one shown in Figure 3.1 trading shares of company d_1 and d_2 plus one shown Figure 3.2 trading shares of company d_3 and d_4. In both Figures, the x-axis show the possible wins of the individual shares and the y-axis shows the probability density of the wins of the shares. The Portfolio Selection Theory assumes that analysts can estimate the three components, the expected wins, their variance and the correlation of distributions of the future wins of the companies. In both figures, the expected win is denoted by μ and the variance is implicitly shown by the width of the Gaussian distribution.

The theory now states, if the win of the companies are uncorrelated and an investor is risk neutral (b=0) all available budget would be invested in d_2 and d_4 in the respective markets, because they have the highest expected win (p_2=1, p_1=0 and $p_4 = 1, p_3$=0)[1]. Furthermore, Figure 3.1 shows the scenario the case where an investor is risk-loving ($b < 0$). The region denoted by "Opportunity for d_1" visualizes why the investor would increase the percentage p_1 of share d_1 in disadvantage of the percentage p_2 of document d_2: by increasing p_1 the variance of the overall win increases which increases the objective function in Equation 3.2 for a risk-loving investor. Furthermore, Figure 3.2 shows the scenario where an investor is risk-averse ($b > 0$). The region denoted by "Risk for d_4" visualizes why the investor would decrease the percentage p_4 of document d_4: the variance of the overall win decreases

[1] This strategy is identified as unreasonable in finance by Markowitz

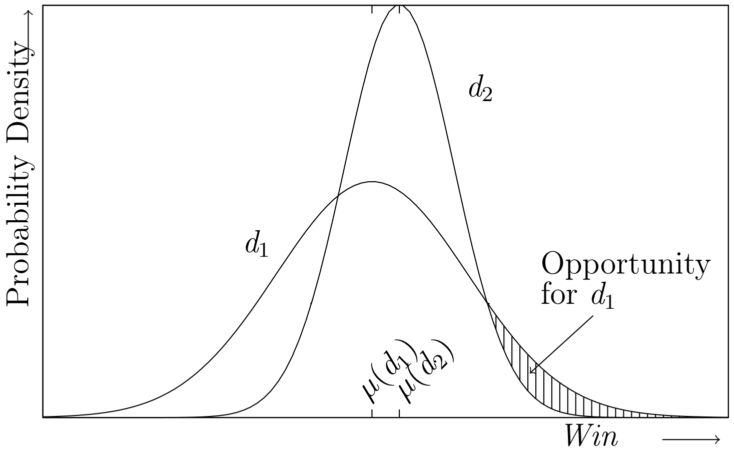


Figure 3.1: *The distribution of the win of the shares d_1 and d_2. The area marked as "Opportunity for d_1" visualizes the reason why a risk-loving investor ($b < 0$), could buy shares of d_1 ($\mu(d) = E[Win(d)]$ is the expected win and the variance of the win is implicitly specified by the shape of the Gaussian).*

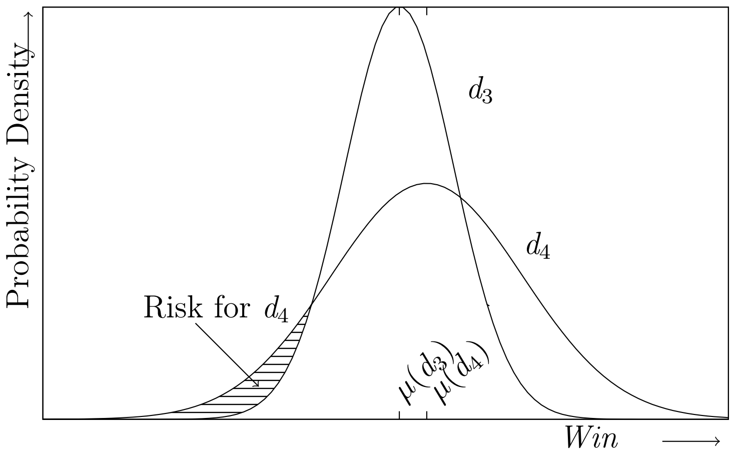


Figure 3.2: *The distribution of the win of the shares d_3 and d_4. The area marked as "Risk for d_4" visualizes the reason why a risk-averse investor ($b > 0$), could buy shares of d_3 ($\mu(d) = E[Win(d)]$ is the expected win and the variance of the win is implicitly specified by the shape of the Gaussian).*

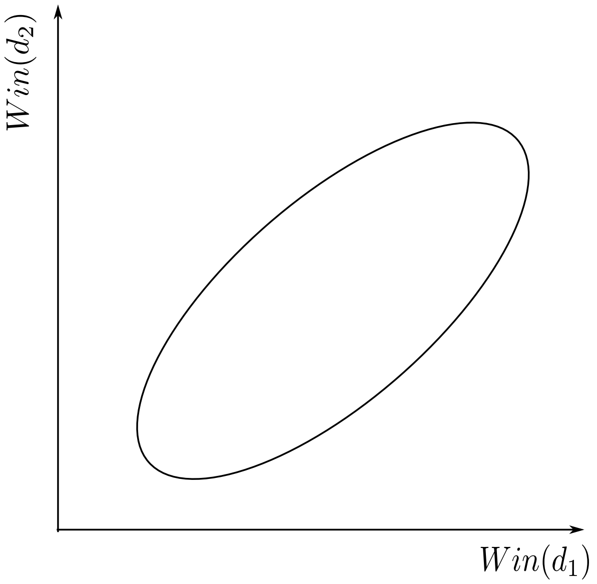


Figure 3.3: *Co-variance between two positively correlated shares d_1 and d_2.*

with an increasing percentage of p_3 which increases the objective function in Equation 3.2 for a risk-averse investor.

Figure 3.3 shows the influence of a correlation of the wins of a company d_1 and company d_2 by a surface plot of the joint Gaussian distribution of the win of the two shares. One can see that the two shares are positively correlated ($cov[Win(d_1), Win(d_2)] > 0$). Intuitively, even though the expected win of company d_1 could be higher if the success of two companies d_1 and d_2 is positively correlated, a risk-averse analyst should invest less in the share d_2 than he normally would (under absence of d_1), because the risk of the overall win increases with investing in both shares. The inverse holds if the analysts have a risk-loving attitude.

3.2.2 Mean-Variance Analysis

Wang (2009) proposes the Mean-Variance Analysis which transfers the selection problem of the Portfolio Selection Theory into a document ranking problem. Here, a document is treated as a share and the uncertain correct score $S'(d)$ of a document d from a text retrieval engine is equivalent to the win of a share d, $Win(d)$. Analogously to the Portfolio Selection Theory this method requires estimates for the expectation, the variance and the co-variances of the uncertain scores. For a probabilistic score function, a posterior probability of relevance given a document representation $\vec{f}$, $P(R|\vec{f})$, the expected score is assumed to be equal to the result of this score function $E[S'(d)] = P(R|\vec{f})$. Wang (2009) and Wang and Zhu (2009) present methods to set the variance and co-variance for different retrieval functions.

In order to transfer the problem from a share selection problem to into a document ranking problem, instead of optimizing the percentages $(p_1, \ldots, p_N)$, the percentages are fixed to the rank j rather than a document (share). Therefore, it is no longer a question anymore, what percentages are invested in each document, but which document to place at which rank and p_j describes the value of rank j. It is not explicitly mentioned how the rank values are set. Wang (2009) shows that the following optimization problem is equivalent to the one from Markowitz in Equation 3.2:

$$\begin{aligned}(d_1, ..., d_N)^* &= \underset{(d_1,...,d_N)}{\operatorname{argmax}} \sum_{j=1}^{N} p_j \; E[S'(d_j)] \\ &- \; b \left[\left[\sum_{j=1}^{N} p_j^2 \operatorname{var}[S'(d_j)] \right] \right. \\ &+ \; \left. \left[\sum_{j=1}^{N} \sum_{k=1,k\neq j}^{N} p_j \; p_k \operatorname{cov}[S'(d_j), S'(d_k)] \right] \right]\end{aligned}$$

with

$$p_1 \geq p_2 \geq ... \geq p_N \text{ with } p_j \in [0:1] \;\; \forall j \qquad \sum_{j=1}^{N} p_j = 1$$

Here, $(d_1, \ldots, d_N)$ is one of $N!$ possible permutations of the documents in $\mathcal{D}$. However, because this optimization problem is computationally expensive, Wang (2009) proposed a greedy algorithm which states that a document d^* should be ranked at position j if it optimizes the following expression:

$$\begin{aligned}d^* &= \underset{d}{\operatorname{argmax}} \, E[S'(d)] \\ &- \; b \; p_j \operatorname{var}[S'(d)] \\ &- \; 2b \sum_{k=1}^{j-1} p_k \operatorname{cov}[S'(d), S'(d_k)]\end{aligned} \tag{3.3}$$

A similar ranking algorithm is used in the presented URR framework.

3.3 The Uncertain Representation Ranking Framework

This section describes the URR framework for ranking documents under uncertain concept-based document representation, which is the main theoretical contribution of this thesis. Similar to the Mean-Variance framework (Wang, 2009), it is based on the Portfolio Selection Theory Markowitz (1952) and uses similar ranking criteria.

3.3.1 Parallels to the Portfolio Selection Theory

This section establishes an intuition of the parallels between the presented URR framework and the Portfolio Selection Theory. This is done by in-

Term in Finance	Term in the URR Framework
Portfolio Selection Theory	URR framework
Share d	Document d
Uncertain win of share $Win(d)$	Uncertain Score $S(d)$
Investment Percentage p_j	Rank value of rank j
Analyst	Divided in two in the URR framework:
Event Analyst	Concept Detector
Senior Analyst	Ranking function
Selection $p_1, ..., p_N$	Ranking $d_1, ..., d_N$

Table 3.1: *Parallels between terms of the URR framework and the Portfolio Selection Theory.*

troducing the most important aspects of the presented URR framework by proposing parallels to terms of the more intuitive terminology of finance. A summary of the parallels is shown in Table 3.1.

Note, despite the similarity, the URR framework is not a one-to-one translation of the Portfolio Selection Theory. The essential difference is how the ranking components, the expected score, its variance and the co-variance are modeled which will be formally explained for the URR framework in Section 3.3.4.

Equivalently to the Mean-Variance Analysis framework, a document in the URR framework corresponds to a share in the Portfolio Selection Theory. Let us assume a set of N shares of companies which exploit natural resources which correspond to the collection of documents $\mathcal{D}$. Each company has one exploitation site for one of n resources, for example oil, gas and uranium. Let the resources have an arbitrary fixed order. Now, let us assume the following event C_i for resource i of company d in the future would be known:

$$c_i(d) = \begin{cases} 1 & \text{the resource } i \text{ will } \textit{not} \text{ run out for company } d; \\ 0 & \text{it will run out..} \end{cases}$$

In the URR framework, the event C_i corresponds to a concept and the fact that a resource does *not* run out for a company corresponds to the concept occurrence of concept i in a document d, c_i. To analyze the future win of the shares, a group of analysts represent companies by these events and the representation of a company is $\vec{C} = (C_1, ..., C_n)$, which corresponds to a document representation in concept-based retrieval consisting in this case of binary concept occurrences.

Now, a *senior analyst* defines a win function of *any* share d, $win(\vec{c})$, by its representation $\vec{c}$. For example, the senior analyst assigns to each type of field i a positive profit $\mathit{profit}_i(1)$ if the resource does *not* run out (c_i=1) and a zero profit $\mathit{profit}_i(0) = 0$ if it does ($c_i = 0$). Then, an obvious definition of the *win* function of a company d would be the following:

$$win(\vec{c}) = \sum_{i=1}^{n} profit_i(c_i) \tag{3.4}$$

The parallel of the *win* function in the URR framework is a score function $score_q$ which calculates the score of *any* document based on its known document representation, the concept occurrences, and the profit function can be imagined as the weights to the score function.

However, the fact that a resource will run out in the future is of course unknown and there are 2^n combinations of resources running out or not, each combination resulting in a particular win. Now, let us assume that for each resource type i there is an *event analyst* who performs tests at the resource i of the company d, which results in a measurement o_i, for example test drills in an oil field. Furthermore, through past experience the analyst can define the probability "resource of type i with measurement o_i will *not* run out." The event analyst corresponds to a concept detector, the measurement to a confidence score and the probability to the probability measure of a concept detector, see Equation 2.2.

With the probabilistic knowledge of each resource of a company, we can calculate a probability distribution over each of the 2^n combinations of resources running out or not occuring in the future. Therefore, we also have a probability distribution of possible wins in the future, with an expected win and its variance. Furthermore, the events of resources running out for different companies could be correlated. For example, if two companies exploit oil at the same oil field, the events whether a resource runs out are strongly correlated and therefore also the win of the companies. This results in the co-variance between wins and so is the win. As a result, the three components of the Portfolio Selection Theory, the expected win, its variance and the co-variance between two wins are defined. The Portfolio Selection Theory would now select investment percentages of the N companies using the estimated win by the senior analyst (Equation 3.4) and the probability distribution of resources running out created by the event analysts. Similarly, the URR framework, which is described in this section, uses a text retrieval functions and the probability distribution over concept occurrences to rank the documents in $\mathcal{D}$. Here, the senior analyst is the parallel to the text retrieval model which will be re-used for concept-based document representations in this thesis.

3.3.2 A Model for Document Representation Uncertainty

In this section a probabilistic model for uncertain document representations and the resulting score is introduced. With this uncertainty model the following section defines the expected score and the score variance which are used to rank documents in parallel to the Portfolio Selection Theory.

In this thesis, the following two document representations and retrieval functions are used. First, the probability of relevance retrieval function using binary representation of concept occurrences in Chapter 5. Second, a language modeling retrieval function using concept frequencies in Chapter 6. In order to be applicable to both cases, the URR framework is formulated for an abstract text retrieval model M with a selection and weighting method and a retrieval function. For a particular query, the selection and weighting method yields a document representation $\vec{F}$ and a score function $score_q$, which takes document representations as an argument, see Section 2.2.1. However, since the document representation is uncertain the following document specific random variable[2] is introduced.

Definition 3.1. Let $\vec{F}(d)$ be the random variable "the document representation of document d".

The distribution of $\vec{F}(d)$ given $\vec{o}$ is the posterior probability after observing the confidence scores $\vec{o}$, $P_\Omega(\vec{F}(d)|\vec{o})$. Furthermore, because of the uncertain document representation, the score of the document is also uncertain. Therefore, we define another document specific random variable.

Definition 3.2. Let $S(d){=}score_q(\vec{F}(d))$ be the "score of a document d which results from the application of the score function $score_q$ on the document representation of document d".

Therefore, $S(d)$ is one of the possible scores of document d which are defined as follows.

$$dom(S(d)) = \{score_q(\vec{f})|\vec{f} \in dom(\vec{F}(d))\}$$

Furthermore, each score $s \in dom(S(d))$ has a probability of being the correct score given the confidence scores $\vec{o}$:

$$P_\Omega(S(d) = s|\vec{o}) = \sum_{\vec{f} \in dom(\vec{F}(d)), score_q(\vec{f})=s} P_\Omega(\vec{F}(d) = \vec{f}|\vec{o})$$

Section 3.3.3 calculates the ranking components of the URR framework based on the distribution of $S(d)$, the expected score and its variance.

3.3.3 Ranking Components

Using the probabilistic model of a document representation and its score from Section 3.3.2, the ranking components of the URR framework, the expected score and its variance, are defined.

[2] For an explanation of document specific document variables see Appendix A.

Expected Score The most important component of the URR framework is the expected score of a document d, $E[S(d)|\vec{o}]$. As $S(d)$ is a function of $\vec{F}(d)$, the expected score can be directly calculated by using the distribution of $\vec{F}(d)$ given the confidence scores of the document (Papoulis, 1984):

$$E[S(d)|\vec{o}] = \sum_{\vec{f} \in dom(\vec{F}(d))} score_q(\vec{f}) P_\Omega(\vec{F}(d) = \vec{f}|\vec{o}) \tag{3.5}$$

Note, this calculation requires the score function, the posterior probability measure and the confidence score of the document to be fully defined.

Variance of the Score Furthermore, the variance of the score $\text{var}[S(d)|\vec{o}]$ given the confidence scores $\vec{o}$ can be calculated as follows:

$$\text{var}[S(d)|\vec{o}] = E[S(d)^2|\vec{o}] - E[S(d)|\vec{o}]^2 \tag{3.6}$$

with

$$E[S(d)^2|\vec{o}] = \sum_{\vec{f} \in dom(\vec{F}(d))} score_q(\vec{f})^2 P_\Omega(\vec{F}(d) = \vec{f}|\vec{o}) \tag{3.7}$$

Here, $E[S(d)^2|\vec{o}]$ is the expected square of the score, and the variance $\text{var}[S(d)|\vec{o}]$ can be expressed as the expectation of the squared score minus the square of the expected score, defined in Equation 3.5. Therefore, the calculation of the variance also requires the definition of the score function, the posterior probability measure and the confidence score to be defined; equivalently to the expected score.

Co-variance of Scores There are good reasons why the scores of two documents could be correlated. As the scores directly depend on the document representation the reason must be that the representations are dependent on each other. An example of such a dependency is given: as the shots of a video are not placed in a random order but follow the story of the video, the occurrence of concepts in adjacent shots will often depend on each other (the fact that a *Person* occurs in a shot could influence the probability that a *Person* occurs in the following shot). This relationship was first explored by work of Yang and Hauptmann (2006) which uses temporal smoothing and hidden Markov models to increase concept detector performance. While this approach is promising, only an oracle model, which is trained on the concept occurrence of the considered collection, was able to achieve significant improvements. Furthermore, no other work in the literature provided a probabilistic model of shots dependencies. As a result, although the use of such dependencies is promising the exploitation of the co-variance for ranking multimedia retrieval is left to future work.

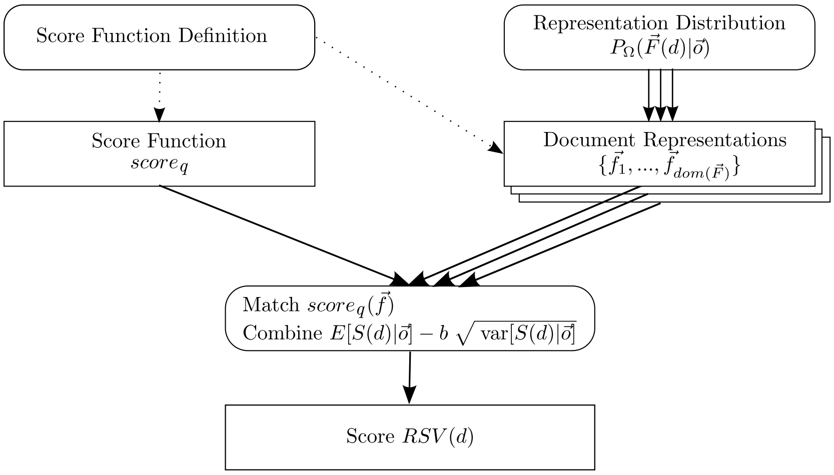

Score Function Definition
Representation Distribution $P_\Omega(\vec{F}(d)|\vec{o})$
Score Function $score_q$
Document Representations $\{\vec{f}_1, ..., \vec{f}_{dom(\vec{F})}\}$
Match $score_q(\vec{f})$
Combine $E[S(d)|\vec{o}] - b\sqrt{\text{var}[S(d)|\vec{o}]}$
Score $RSV(d)$

Figure 3.4: *Extract of modified information retrieval process, see Figure 1.2 for the full process.*

3.3.4 Combining the Components

In the following, the proposed way to combine the ranking components in the URR framework is described, which is derived from the Portfolio Selection Theory in parallel to the greedy algorithm of the Mean-Variance Analysis framework (Wang, 2009) shown in Equation 3.3. However, due to the absence of the co-variances, which was described in the previous section, the ranking framework can be formulated as a closed formula (not depending on the previously ranked documents):

$$RSV(d) = \underbrace{E[S(d)|\vec{o}]}_{\text{Eq. 3.5}} - b\underbrace{\sqrt{\text{var}[S(d)|\vec{o}]}}_{\text{Eq. 3.6}} \tag{3.8}$$

Here, $\sqrt{\text{var}[S(d)|\vec{o}]}$ is the standard deviation of the score. The ranking score value, $RSV(d)$, in Equation 3.8, is the value by which a document is ranked in the URR framework.

In fact, Equation 3.8 is a mathematical formulation of the *Combine* step from the modified information retrieval process, proposed in Section 1.4. Figure 3.4 makes the proposed modification of the information retrieval process explicit: there are $|dom(\vec{F})|$ different possible document representations of each document and the probability $P_\Omega(\vec{F}(d) = \vec{f}|\vec{o})$ specifies the probability of a particular representation $\vec{f}$ to being the correct representation of document d. Afterwards, the expected score and the variance are combined to the final ranking score value for the document.

Difference to the Mean-Variance Analysis Framework There are the following differences to the Mean-Variance Analysis framework. First, the Mean-Variance Analysis framework is restricted to probabilistic ranking functions, which is not the case in the URR framework. Second, the standard deviation of the score is used instead of the variance to represent the risk or opportunities of a retrieval engine. The reason for this is given in Section 3.3.5. Finally, because the URR framework concentrates on the effects of the document representation uncertainty, the "rank values" p_j for a document at rank j are assumed to be constant and therefore can be ignored in the calculation of the ranking score value. This is similar to having constant cost for reading a document, which was for example also used in the Probability of Relevance Ranking Principle (Robertson, 1977) to show its optimality.

Differences to the Portfolio Selection Theory There are the following differences between the URR framework and the Portfolio Selection Theory. First, because the percentages are constant and fixed to a rank and not to a document, similarly to the Mean-Variance Analysis framework, the risk setting of $b = 0$ is not an unreasonable choice in the URR framework. On the contrary, it intuitively appears reasonable because it ranks the documents by their expected scores which is a good estimator for $score_q(\vec{f})$. Therefore, the standard deviation only adds something on top of an already reasonable solution rather than making the combination reasonable, which is the case in the Portfolio Selection Theory.

Implementation Algorithm 3.1 shows an exemplary implementation of the URR framework proposed in Section 1.4. The employed retrieval model M is based on a document representation $\vec{F}$, which are later either concept occurrences or concept frequencies. For each document, the ranking score value is not calculated directly but first the expected score and the expectation of the squared score are calculated by calling the score function on all possible document representations. Finally, the score is combined according to Equation 3.8.

3.3.5 Efficient Implementation and Practical Considerations

Monte Carlo Sampling The number of possible document representations $|dom(\vec{F})|$ is often large and calculating the components of Equations 3.8 can be computationally expensive. In such situations it is possible to use the Monte Carlo Sampling method, which was designed to reduce the costs of calculating computationally expensive expectations (Liu, 2002). The method can be applied as follows: let there be NS random samples $\vec{f}(d^1), ..., \vec{f}(d^{NS})$ from the distribution of possible representations $P_\Omega(\vec{F}(d)|\vec{o})$ for document d (The way the samples are created is document representation dependent and an example is described in Chapter 6). The expectations from Equation 3.5 and expectation of the squared score needed for the calculation of

Algorithm 3.1: Proposed modification to the information retrieval process re-using a text retrieval model M.

```
Data: Collection $\mathcal{D}$,
Query Features $\vec{QF}$,
Risk Parameter $b$,
Retrieval model $M = (selectNweight_M(), retfunc_M\langle\rangle)$,
Distributions $P(\vec{F}(d)|\vec{o})\forall d$.

$retrievalrun(\vec{qf} : dom(\vec{QF}))$
begin
    // Score Function Definition
    $(\vec{F}, w) := selectNweight_M(\vec{qf})$
    $score_q := new\ retfunc_M\langle\vec{F}, w\rangle$
    // Matching and Combine
    foreach $Document\ d\ in\ \mathcal{D}$ do
        $ES := 0$ // $E[S(d)|\vec{o}]$ ;
        $ES2 := 0$ // $E[S(d)^2|\vec{o}]$ ;
        // Calculate $ES$ and $ES2$ according to Eq. 3.5 and Eq. 3.7
        foreach $Representation\ \vec{f} \in dom(\vec{F}(d))$ do
            $s = score_q(\vec{v})$;
            $ES{+}{=}s\ P(\vec{F}(d) = \vec{f}|\vec{o})$;
            $ES2{+}{=} + s^2\ P(\vec{F}(d) = \vec{f}|\vec{o})$;
        end
        // Combine according to Equation 3.8
        Append $(d, ES - b\ \sqrt{ES2 - ES^2})$ to $ranking$
    end
    return $sort(ranking, ranking.score\ DESC)$
end
```

the variance in Equation 3.6, can then be approximated by:

$$E[S(d)|\vec{o}] \simeq \frac{1}{NS}\sum_{l=1}^{NS} score_q(\vec{f}(d^l)) \tag{3.9}$$

$$E[S(d)^2|\vec{o}] \simeq \frac{1}{NS}\sum_{l=1}^{NS} score_q(\vec{f}(d^l))^2 \tag{3.10}$$

Here, the expected score and the expected squared score are calculated by the sum of the scores of the samples divided by the number of samples. This calculation has a linear run-time complexity in the number of samples. Because the standard error of the Monte Carlo estimate is in the order of $1/\sqrt{NS}$ a good estimate is already achieved with relatively few samples. Note, that there are more advanced sampling methods, which essentially reduce the required samples. For example, importance sampling (Liu, 2002) can be used to prefer rare representations during sampling and correct the

Algorithm 3.2: More efficient, approximate implementation of the retrieval system in Algorithm 3.1.

```
**Data**: Collection $\mathcal{D}$,
Collection of samples $\mathcal{D}_S$,
Query Representation $\vec{QF}$,
Document Features $\mathcal{V}$,
Risk Parameter $b$,
Retrieval model M=($selectNweight_M(), retfunc_M\langle\rangle$),
Distribution $P(\mathcal{V}|\vec{o})$,
Number of samples $NS$

$GenerateSamples()$
begin
    foreach $Document\ d\ in\ \mathcal{D}$ do
        for $l$=1 to $NS$ do
            $d^l := newDoc()$;
            $\mathcal{V}(d^l) :=$ sample from $P(\mathcal{V}|\vec{o})$;
            Append $d^l$ to $\mathcal{D}_S$;
        end
    end
end

$retrievalrun(\vec{qf} : dom(\vec{QF}))$
begin
    // Score Function Definition
    $(\vec{F}, w) := selectNweight_M(\vec{qf})$
    $score_q := new\ retfunc_M\langle\vec{F}, w\rangle$
    // Matching and Combine
    foreach $Document\ d\ in\ \mathcal{D}$ do
        $ES' := 0$ // $\simeq E[S(d)|\vec{o}]$ ;
        $ES2' := 0$ // $\simeq E[S(d)^2|\vec{o}]$;
        foreach $Sample\ Document\ d^* \in \mathcal{D}_S\ of\ d$ do
            $s = score_q(\vec{f}(d^*))$;
            $ES' += s$;
            $ES2' += s^2$;
        end
        $ES' = ES'/NS$;
        $ES2' = ES2'/NS$;
        Append $(d, ES' - b\sqrt{ES2' - ES^2})$ to $ranking$;
    end
    return $sort(ranking, ranking.score\ DESC)$
end
```

resulting bias via a weighting scheme. However, this thesis focuses on the qualitative results of sampling and leaves more advanced sampling methods for future work.

Algorithm 3.2 describes the implementation of the sampling alternative of the URR framework. Here, at indexing time the procedure *GenerateSamples* generates a fixed number of NS samples from the distribution of the feature vocabulary $\mathcal{V}$. The resulting artificial documents are stored in the collection $\mathcal{D}_S$. At retrieval time, for each document d in collection $\mathcal{D}$ the retrieval score and the squared retrieval score are summed in the variables ES' and $ES2'$ respectively. Afterwards, both variables are divided by the number of samples NS and ES' is the estimate $E[S(d)|\vec{o}]$, see Equation 3.9, and $ES2'$ is the estimate $E[S(d)^2|\vec{o}]$, see Equation 3.10. A document is then ranked by $ES' - b\sqrt{ES2' - ES^2}$, which is equal to the ranking objective of the URR framework in Equation 3.8.

Collection Prior Estimation Many probabilistic retrieval functions, including the two retrieval functions proposed in Chapter 5 and Chapter 6, use a prior of the concept occurrence in the collection $\mathcal{D}$ as a feature statistic. For a concept C the prior in a collection with N documents can be defined as, see Appendix A:

$$P(C) = \frac{\sum_{j=1}^{N} c(d_j)}{N}$$

Because the concept occurrences are unknown, the prior is also unknown. However, it can be estimated by its expected value:

$$E[P(C)|o(d_1), ..., o(d_N)] = \frac{\sum_{j=1}^{N} E[C(d_j)|o(d_j)]}{N} = \frac{\sum_{j=1}^{N} P_\Omega(C(d_j)|o(d_j))}{N} \quad (3.11)$$

Here, a similar derivation to the expected win in the Portfolio Selection Theory in Equation 3.1 is used. This estimation will be used throughout this thesis and for brevity will be denoted by just $P(C)$.

Requirements for the Score Function Croft (1981) was the first to propose ranking by the expected score for probabilistic indexing by ranking by the expected score of the binary independence weight as defined in Section 2.4.4. However, as reported by Fuhr (1989), this function does not calculate a score which can be expected to be rank equivalent with the expected probability of relevance score function.

The URR framework also uses the expectation and variance of a score function. Therefore, the reasons for this phenomenon are investigated because they present limitations to the score functions which are applicable to the framework: In order to efficiently rank documents according to a score function which is derived from a theoretically motivated retrieval function, $score_q$, retrieval engines often use a simplified score function, $score'_q$, which produces rank-preserving scores. However, as found by Fuhr (1989), these simplifications cannot always be expected to be rank preserving when calculating the expected scores. In other words, the following does not always hold:

$$E[score_q(\vec{F}(d))] \propto E[score'_q(\vec{F}(d))]$$

In the following, the most common simplifications are analyzed to see if they can be assumed to be rank preserving. A typical example of such simplifications is the binary independence model (Robertson et al., 1982), which is translated to concept occurrences here:

$$\begin{aligned} P(R|C = \vec{c}) &\propto \frac{P(\vec{C} = \vec{c}|R)}{P(\vec{C} = \vec{c})} && (3.12)\\ &\propto \frac{P(\vec{C} = \vec{c}|R)}{P(\vec{C} = \vec{c}|\bar{R})} && (3.13)\\ &\propto \sum_c \log\left(\frac{P(C = \vec{c}|R)}{P(\vec{C} = \vec{c}|\bar{R})}\right) && (3.14)\end{aligned}$$

Now, the simplifications on the rank equivalence of the above simplifications are discussed:

(1) In Equation 3.12, after the Bayesian inversion, the query-specific relevance prior $P(R)$ is left out in the simplified score function, $score'_q$, which is common practice in information retrieval. From the laws of expectations, see Appendix A, the following holds:

$$E[P(R|\vec{C} = C(d)|\vec{o}] = P(R)\; E\left[\frac{P(\vec{C} = C(d)|R)}{P(\vec{C} = C(d))}\middle|\vec{o}\right]$$

$$\text{var}[P(R|\vec{C} = C(d))|\vec{o}] = P(R)^2\; \text{var}\left[\frac{P(\vec{C} = C(d)|R)}{P(\vec{C} = C(d))}\middle|\vec{o}\right]$$

Therefore, the expected value of $score'_q$ is linearly proportional to the expected original $score_q$. However, the calculation of the variance depends quadratically on the information need specific constant $P(R)$. As a result, when using the simplified score function $score'_q$ the risk factor b from Equation 3.8 has a different influence on queries with a high relevance prior than on queries with a low relevance prior. This is probably the reason why the parameter b is easier to control by using the standard deviation in Equation 3.8 which only depends linearly on $P(R)$ but still expresses the risks and opportunities.

(2) In Equation 3.13 the odds are calculated instead of the original probability and in Equation 3.14 a product is transformed into a sum of logarithms. Both simplifications are not linear transformations of the

score function $score_q$. And it is easy to find two documents for which the following does not hold:

$$E[P(R|\vec{C} = C(d)|\vec{o}] \not\propto E[log(O(R|\vec{C} = C(d')))|\vec{o})]$$

Therefore, the expected score of $score'_q$ is also not always proportional to the expectation of the original function $score_q$. Therefore, these two kinds of transformations should not be used to simplify the original score function $score_q$.

To summarize, among the common simplifications of ignoring constants, ranking by the odds or the log of the score functions only ignoring constants preserves the ranking of the expected score.

3.4 Summary and Discussion

This chapter presented the generic Uncertain Representation Ranking (URR) framework based on uncertain concept-based document representations. The framework allows the use of existing text retrieval functions in concept-based retrieval. The URR framework was derived from the Portfolio Selection Theory (Markowitz, 1952). The parallels were highlighted by an analogy between the financial setting of the Portfolio Selection Theory and the URR framework. The main difference to the original theory is that a ranking of documents is created rather than selecting optimal investment percentages of shares (a portfolio). The mathematical derivation of the ranking function of the URR framework was done in parallel to the Mean-Variance Analysis framework Wang (2009), which models the uncertainty of scores in text retrieval.

Furthermore, the URR framework extends the Portfolio Selection Theory by explicitly modeling *why* the score of a document has a certain distribution. The Portfolio Selection Theory and the Mean-Variance Analysis framework only assume that the expected win of a share, its variance and the co-variances between shares *are known* (or equivalently for scores in the case of the Mean-Variance framework). On the other hand, the URR framework assumes that the score depends on the concept-based document representation, which is uncertain. As a result, the score in the URR framework is *only* uncertain because of the uncertainty of the concept-based document representation, which in turn is probabilistically modeled and the expected score and its variance can be determined.

The expected score and its standard deviation (the square root of the variance) are the two ranking components in the URR framework. The components are combined using a risk factor which represents the risk attitude of the retrieval engine. Furthermore, as the number of possible document representations can be large a method of calculating the expected score and its variance by means of Monte Carlo Sampling is provided. Finally, as most retrieval systems use a simplified score function compared to the theoretical proposed score function, we have analyzed which commonly used simplifications are rank-preserving and can therefore be used in the URR framework.

The URR framework will be evaluated for representations of binary concept occurrences in video shot retrieval in Chapter 5 and for representations of concept frequencies in video segment retrieval in Chapter 6.

Chapter 4

Concept Selection and Weighting

This chapter is based on Aly et al. (2009).

4.1 Introduction

Today, virtually all queries to retrieval engines are formulated in text queries[1]. Therefore, this chapter presents a method that selects good concepts from a textual query to be used for concept-based retrieval.

The selection of good concepts for an information need is more difficult than selecting good terms in text retrieval. In the latter, document features and query features correspond and this can be used to select the features to be used in a document representation. For example, the occurrence of the term 'President' in a textual query is normally used to select the term frequency feature of this term for searching. However, in order to select concepts from textual queries for concept-based retrieval, there are the following new challenges:

(1) A mapping between the query text and the available concepts has to be found since many concepts will not directly be named in the query.

(2) A measure for the helpfulness of each concept has to be estimated.

(3) Good concepts have to be defined, according to this measure.

(4) The weights of a score function have to be estimated.

For example, in the information need "President Obama" the concept *US-Flag*, which does not appear in the query text, will often occur in relevant shots. Therefore, using the concept *US-Flag* for searching will help to answer the information need. However, the question is, how should a computer algorithm decide whether a concept is good or not and what weights to assign to it?

[1] David Neal, "Google explains how it ranks pages", the Inquirer (Fri Feb 26 2010)

This chapter proposes the Annotation-Driven Concept Selection (ADCS) method. The ADCS method assumes the existence of a development collection in which all concepts are annotated by humans. Such collections normally exist, since concept detectors need them for training. For this collection, a textual representation for each shot is created. A standard text retrieval engine is used to index the textual representation of development collection. Now, given a query, the concept selection and weighting is executed as follows.

(1) Execution of the original text query on the textual representation of the development collection indexed by the text retrieval engine.

(2) Estimation of a goodness measure and weighting for each concept using the scores returned by the text retrieval engine together with the human annotated concept occurrences.

(3) Selection of the concepts to use in the search according to the goodness measure.

This method is similar to a pseudo relevance feedback method on a different collection. However, for the initial search it uses a different search technique (text retrieval) than in the actual search collection, where concept-based retrieval is performed.

The remainder of this chapter is structured as follows: in Section 4.2 background on measurable concept selection objectives and evaluation approaches are given. Section 4.3 presents the annotation-driven concept selection and weighting method, proposed by this chapter. In the following, Section 4.4 describes the experiments which were carried out to evaluate the quality of the proposed algorithm. Finally, in Section 4.5 the chapter is summarized and the findings are discussed.

4.2 Background: Concepts Selection Objectives and Evaluation

In this section, first selection objectives for good concepts for an information need are presented. Afterwards, existing evaluation methods for concept selections are described.

4.2.1 Mutual Information

The expression "a good concept for an information need" is qualitative and does not suggest *how* to decide, whether a concept is good for an information need. In practice, such decisions are often made by calculating some goodness measure and then either selecting items above a certain threshold or selecting a fixed number of items with the highest measure. Hauptmann et al. (2007)

find that Mutual Information (Arndt, 2001) between relevance and concept occurrence to be a suitable goodness measure for a concept to be selected for a given information need. The Mutual Information is defined as follows:

$$MI(R; C) = \sum_{c,r \in \{0,1\}} P(C = c, R = r) \log \left(\frac{P(C = c, R = r)}{P(C = c)P(R = r)} \right) \tag{4.1}$$

Note, in the original work from Hauptmann et al. (2007) the event space for the probability measure is not specified. In this thesis, we assume that the probability measure is defined on the collection $\mathcal{D}$, because concept selections and weightings should be collection dependent.

4.2.2 Evaluation of Concept Selection and Weighting

Because of its importance in the retrieval process, it is desirable to evaluate the performance of concept selection and weighting methods independently from the final search performance. However, there is little research on the evaluation of such methods. We were the first to propose a concept selection evaluation method based on human judgments (Hauff et al., 2007). Here, humans select concepts which they expect to be important for an information need. The concept selection method which is evaluated is assumed to return a ranked list of concepts. The evaluation is then performed using the mean average precision (MAP), considering concepts as documents and the importance of the concept as its relevance. Since the users were novice to the retrieval domain, the evaluation is aimed at a general applicability of a concept, not whether it is useful in a specific collection.

Huurnink et al. (2008) provide a more extensive study of evaluation measures for concept selection methods. Two so-called benchmarks are defined which are described in the following. First, the *user benchmark* is defined, for which users ordered concepts according to their importance to an information need. Second, the *collection benchmark* is defined. The collection benchmark uses a collection with known concept occurrence and relevance judgments to calculate the Mutual Information between each concept and relevance, see Equation 4.4. The concepts are then ordered by their Mutual Information. Therefore, while the user benchmark contains generally useful concepts, the collection benchmark is collection-specific to the collection, on which the Mutual Information is calculated. For both benchmarks, two evaluation measures are proposed: the *set agreement*, which evaluates whether the concepts selected by a concept selection method overlap with the selection in the benchmark, and *rank correlation*, which measures correlation between a ranking of concepts, returned by a concept selection method, with the ranking defined by the benchmark by Spearman's correlation (Triola, 2008). Currently, there is no evaluation measure which quantifies the goodness of estimated concept weights. Therefore, we limit the evaluation of our concept selection and weighting method to the selected concepts.

4.3 Annotation-Driven Concept Selection

In this section, the ADCS method is described. Ideally, a concept selection should be based on the search collection in order to be collection-specific. However, relevance and concept occurrences are unknown in the search collection which makes the selection of concepts difficult. Therefore, in the ADCS method the selection is done on a development collection which has been completely annotated with the occurrences of all available concepts in the concept vocabulary and only the relevance to a certain information need is uncertain. To attain information about relevance, while giving the user a high degree of freedom to formulate his query, the development collection is textually described and an initial text search is performed on the textual representation of the development collection.

4.3.1 Text Collection from Development Collection

In order to perform a search on the development collection using a textual query, for each shot s a textual description $desc(s)$ is produced. In general, the descriptions should be made in a way that a text retrieval engine can return a good ranking of the documents with regards to the relevance of the underlying shot. Ideally, the text descriptions meet the following criteria. First, they are precise (unambiguous). Second, they are exhaustive, so that all words that a user could use to express his information need will be properly represented. Unfortunately, the two criteria contradict each other since a longer text inevitably introduces more ambiguity.

In the following, the used components of the textual description, $desc(s)$, are described. Clearly, words being said during the shot have a descriptive nature for the shot's content and therefore the output of an automatic speech recognition system, $asr(s)$, is included in the description. However, many information needs will be concerned with the visual content of a shot so that the spoken words will only be of limited help. Therefore, textual descriptions for occurring concepts, which often describe the visual content, are also considered. The ADCS method investigates the following concept descriptions. First, the name and definition of the concept C, $def(C)$. Second, the content of a Wikipedia article about this concept, $wiki(C)$. In order to obtain the article text, an article with the concept's name was automatically downloaded from Wikipedia[2]. If this resulted in a disambiguation page or the page did not exist, an article was chosen manually. Therefore, for a given shot s the textual shot description is the concatenation of the above components:

$$desc(s) = asr(s) + \sum_{C \in \mathcal{V}_C, c(s)=1} [def(C) + wiki(C)]$$

[2] http://en.wikipedia.org

Here, the + operator refers to the concatenation of text. Note, this is only one possible way of creating shot descriptions, and further improvements are left to future work.

After the creation of the text descriptions, the development collection is indexed by a standard text retrieval engine. This index is used at query time to efficiently rank shots through their description. This fulfills the first step of the search procedure which was described in Section 4.1.

4.3.2 Occurrence Probability of a Concept Given Relevance

This section describes how the ADCS method estimates the occurrence probability of a concept given relevance, $P(C|R)$, which will be used to calculate the mutual information of a concept and relevant shots. Furthermore, this probability can be used in several retrieval functions, including the one proposed in Chapter 5. This estimation is the second step in the concept selection procedure described in Section 4.1.

It is assumed that the concept occurrence in relevant shots in the development collection $\mathcal{D}'$ are similarly distributed as in the search collection $\mathcal{D}$. This is a reasonable assumption for video data of the same domain. Therefore we assume:

$$P(C|R) = P_{\mathcal{D}'}(C|R) \text{ for all } C \in \mathcal{V}_C$$

To start, the query is executed on the textual description of the development collection by a text retrieval engine TXT. As a result, for each shot s its concept occurrences $\vec{c}(s)$, its rank and its text retrieval score $score_{TXT}(\vec{f})$ are known. Here, $\vec{f}$ are the features used by the text retrieval engine. Let $s_1, ..., s_N$ be the ranking of all N shots in the development collection $\mathcal{D}'$. In a perfect ranking where the m relevant shots in the development collection are at the top of the ranking, the probability of a concept occurrence in relevant shots can be calculated as follows:

$$P(C|R) \simeq \frac{|\{s_i|c(s_i) = 1, i \leq m\}|}{m} \tag{4.2}$$

Here, the number of relevant shots with concept C is divided by the number of relevant shots m. However, in reality not all relevant shots will be in the highest ranks. At best, the relevant shots will have higher ranks than irrelevant shots. Therefore, m is fixed to an empirical value and is now the cut-off value indicating the shots which are considered for the estimation, similar to Croft and Harper (1979). The probability of a concept occurrence given relevance $P(C|R)$ is then estimated as follows:[3]:

$$P(C|R) \simeq \frac{\sum_{i=1,c(s_i)=1}^{m} score_{TXT}(\vec{f}(s_i))}{\sum_{i=1}^{m} score_{TXT}(\vec{f}(s_i))} \tag{4.3}$$

Here, the sum of all scores of the first m shots in which a concept occurs are divided by the total sum of the scores of the first m shots. Therefore, the estimate is always normalized. This has the effect that shots which occur later in the ranking have less influence on the estimation.

4.3.3 Concept Selection

In this section, the actual method of selecting concepts, based on the previously estimated probability, is described. The Mutual Information of R and a concept C can be calculated with the conditional probability $P(C|R)$ and the two priors $P(C)$ and $P(R)$. The estimation of the first probability is described above. The second can be estimated from the detector output of the search collection, see Section 3.3.5. Therefore, when assuming a small value for the prior probability of relevance the Mutual Information can be estimated. Throughout this thesis, 50 relevant shots per query are assumed and we have: $P(R) = \frac{50}{N}$. Using these probabilities together with the law of total probability, it is possible to estimate the probability of a concept occurring in irrelevant shots:

$$P(C = 1|R = 0) = \frac{P(C = 1) - P(C = 1|R = 1)P(R = 1)}{1 - P(R = 1)}$$

Now, an estimate for the Mutual Information can be given:

$$MI(R; C) = \sum_{c,r\in\{0,1\}} P(C = c|R = r)P(R = r)log\left(\frac{P(C = c|R = r)}{P(C = c)}\right) \tag{4.4}$$

Note, the above calculation follows directly from its definition in Equation 4.4. In the following, the concept vocabulary is sorted using this estimate and a fixed number of n top-ranked concepts is selected for the search in the search collection.

4.3.4 Implementation

Algorithm 4.1 describes a basic implementation of the ADCS method. First, the query (represented by its features) is passed to a text retrieval engine TXT, which returns a ranking l, consisting of pairs of documents and scores.

[3] In Aly et al. (2009) this estimation method was termed *score-based* estimation. Furthermore, a second estimation, the *certainty-based* was proposed. However, the certainty-based method never improved upon the score-based method and therefore only the score-based method is used throughout this thesis.

For the top-m shots in the ranking where the concept C occurs the scores are summed up to the query-specific weight for the concept, $w(C)$, which is the occurrence probability of the concept C in relevant shots, $P(C|R)$. In the following, the weights are normalized by the total sum of the scores from the top-m shots. The list of concepts is then sorted by the Mutual Information and the top-n concepts are returned.

Algorithm 4.1: *The Annotation-Driven Concept Selection method.*

```
Data: Concept Vocabulary $\mathcal{V}_C$,
Query features $\vec{QF}$,
Rank cut-off $m$,
Number of concepts to use in query $n$,
Text retrieval engine TXT,
Concept weights $w$ with $w(C) = P(C|R)$

$selectNweight_{ADCS}(\vec{qf} : dom(\vec{QF}))$ :
begin
    $l := retrievalrun_{TXT}(\vec{qf})$
    $sum := 0$
    for $j = 0; j < m; j{+}{+}$ do
        $(d, s) := l[j]$ // $j$th shot
        $sum{+}{=}s$
        foreach $c \in \mathcal{V}_C$ do
            if $c(d) = 1$ then
                $w_C{+}{=}s$
            end
        end
    end
    // Normalization
    foreach $C \in \mathcal{V}_C$ do
        $w(C)/{=}sum$
        calculate $mi(C)$ // according to Eq. 4.4
        append $C$ to $concepts$ list
    end
    $concepts := sort(concepts, mi(C)\ DESC)$
    // return the top-$n$ concepts sorted by the the mutual information
    return $(top(concepts, n), w)$
end
```

4.4 Experiments

In this section, the ADCS method is evaluated independently of the retrieval performance for a search collection. For information on the influence of the ADCS method on the search performance, the reader is referred to Chapter 5 and Chapter 6.

4.4.1 Experiment Setup

We use the TRECVid 2005 development collection with the official 24 TRECVid 2005 queries[4] for the evaluation. Additionally, we consider the MediaMill vocabulary which comprises 101 concepts (Snoek et al., 2006) and the Vireo subset (Jiang et al., 2007) which comprises 374 concepts of the LSCOM vocabulary (Naphade et al., 2006) together with the corresponding annotations. Note, this setup deviates from the envisioned scenario of the ADCS method, since the evaluation is performed on the same collection as the one which is used for concept selection. However, since there was no other collection available with a sufficient amount of annotations and relevance judgments, this was the only possible experiment setup to evaluate concept selection.

Performance Measures The quality of the proposed concept selection method is evaluated through the average precision of the ranking, see Hauff et al. (2007), and rank correlation of the collection benchmark from Huurnink et al. (2008). For our evaluation method, proposed in Hauff et al. (2007), it is assumed that all concepts with a positive Mutual Information for a query are relevant and the collection benchmark was chosen since the proposed method aims to find concepts for the search collection. The set agreement measure is not reported because it does not contain information of the ordering of concepts.

Baselines We use two baselines for the evaluation: *text matching*, the best performing baseline method from (Huurnink et al., 2008), and *wiki-article*, from our previous work (Hauff et al., 2007). The text matching method matches the concept description with the query text (after stop word removal) using the vector space model. Concepts are then returned in the order of the vector space score. Furthermore, the wiki-article method uses articles from Wikipedia to describe the concepts. A normal text retrieval engine is then used to rank the documents for a certain query.

Initial Text Retrieval Run The initial text retrieval runs were performed with the general purpose retrieval engine PF/Tijah (Hiemstra et al., 2006)

[4] Relevance judgments were kindly provided by Rong Yan formerly at Carnegie Mellon University (Yan and Hauptmann, 2007)

using the NLLR retrieval model (Rode and Hiemstra, 2006) to rank shots in the development collection. The search performance of the shot descriptions based on the Vireo vocabulary was poor, see Section 4.4.2. Therefore, we also evaluated the concept selections of the Vireo vocabulary based on the results of the initial text retrieval run on the text description from the MediaMill vocabulary for comparison. Note that it is possible to use different shot descriptions, for example using the MediaMill vocabulary, to generate a ranking of shots, but perform the weight estimation from Equation 4.3 and Equation 4.4 for another concept vocabulary.

Result Presentation All results are displayed using quartiles, visualizing the distribution of the performance measure of the 24 queries. The elements of this distribution are defined as follows: let q be a query and $pm(q)$ be the performance measure of this query. Furthermore, let $(q_1, \ldots, q_N)$ be the sorted list of the queries by their performance measure pm. The lowest point is the lower outlier $m(q_1)$. The first quartile is the point dividing the lowest quarter of the queries $pm(q_{\lfloor \frac{N}{4} \rfloor})$ from the rest. The second quartile (the median) is the performance of the middle query $pm(q_{\lfloor \frac{N}{2} \rfloor})$. The third quartile is the point, dividing the upper quarter of the queries $pm(q_{\lfloor \frac{3N}{4} \rfloor})$ from the rest. Finally, the highest point is the upper outlier $pm(q_N)$.

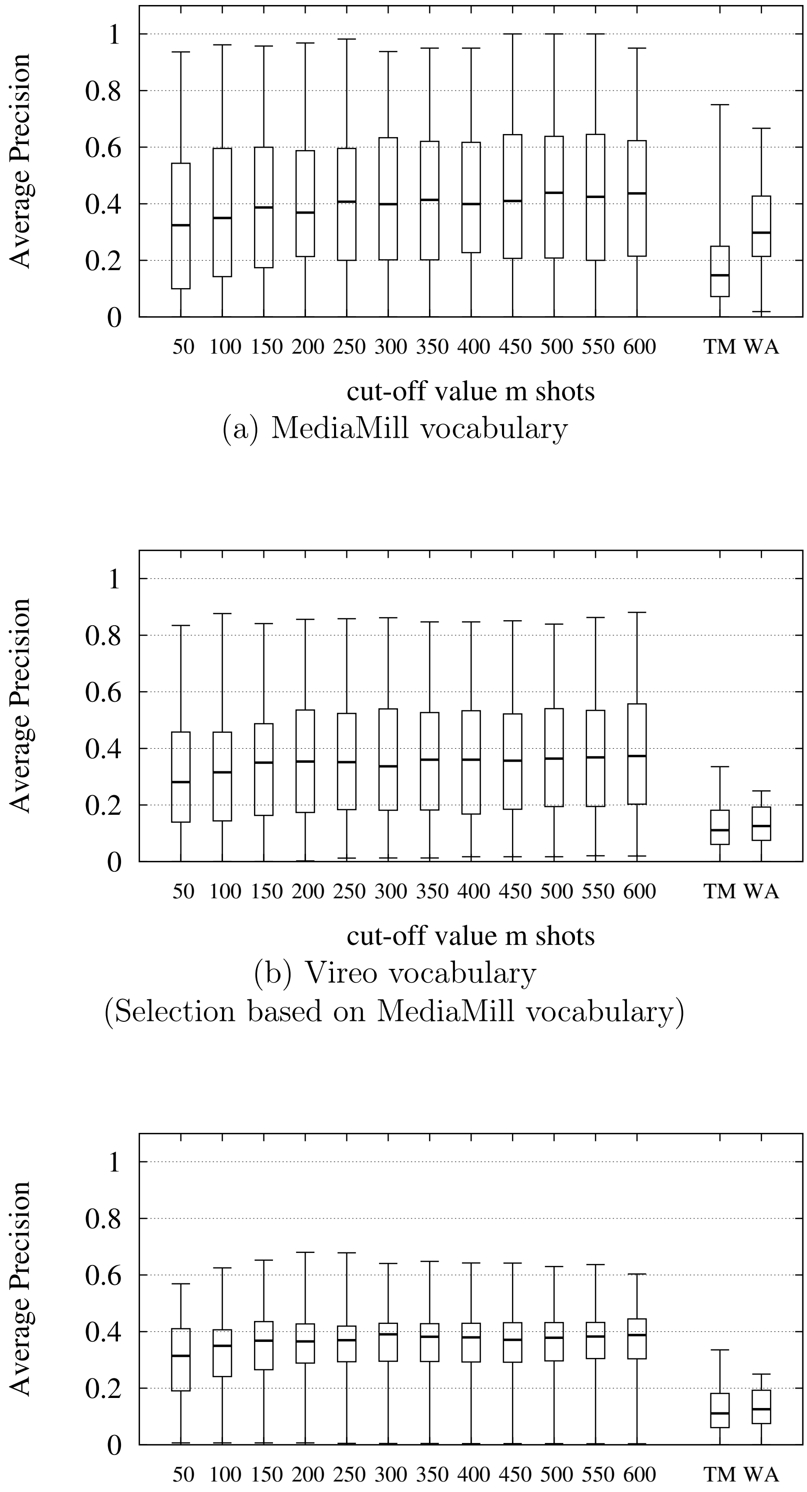
4.4.2 Initial Text Retrieval Run

The proposed method depends on a good precision of the initial text retrieval run. Therefore, as a preliminary indicator of the selection performance the MAP of the 24 TRECVid 2005 queries on the development collection was evaluated. For the shot descriptions using the MediaMill vocabulary, this resulted in a search performance of 0.26 MAP. On the other hand, the shot descriptions using the Vireo vocabulary only resulted in a search performance of 0.14 MAP.

4.4.3 Evaluation of Concept Selection

This section describes the experiments which were performed to assess the effectiveness of the proposed concept selection and weighting method. Figure 4.1 and Figure 4.2 show the comparison of the ADCS method with the *text matching* baseline method from (Huurnink et al., 2008) and our wiki-article baseline (Hauff et al., 2007). The performance measures are reported separately for the MediaMill and the Vireo vocabularies. In all plots, the x-axis denotes the different cut-off values m with the two baselines shown on the right.

Average Precision Figure 4.1 shows the results of the average precision measure for the two concept vocabularies. The y-axis shows the *average*



(a) MediaMill vocabulary

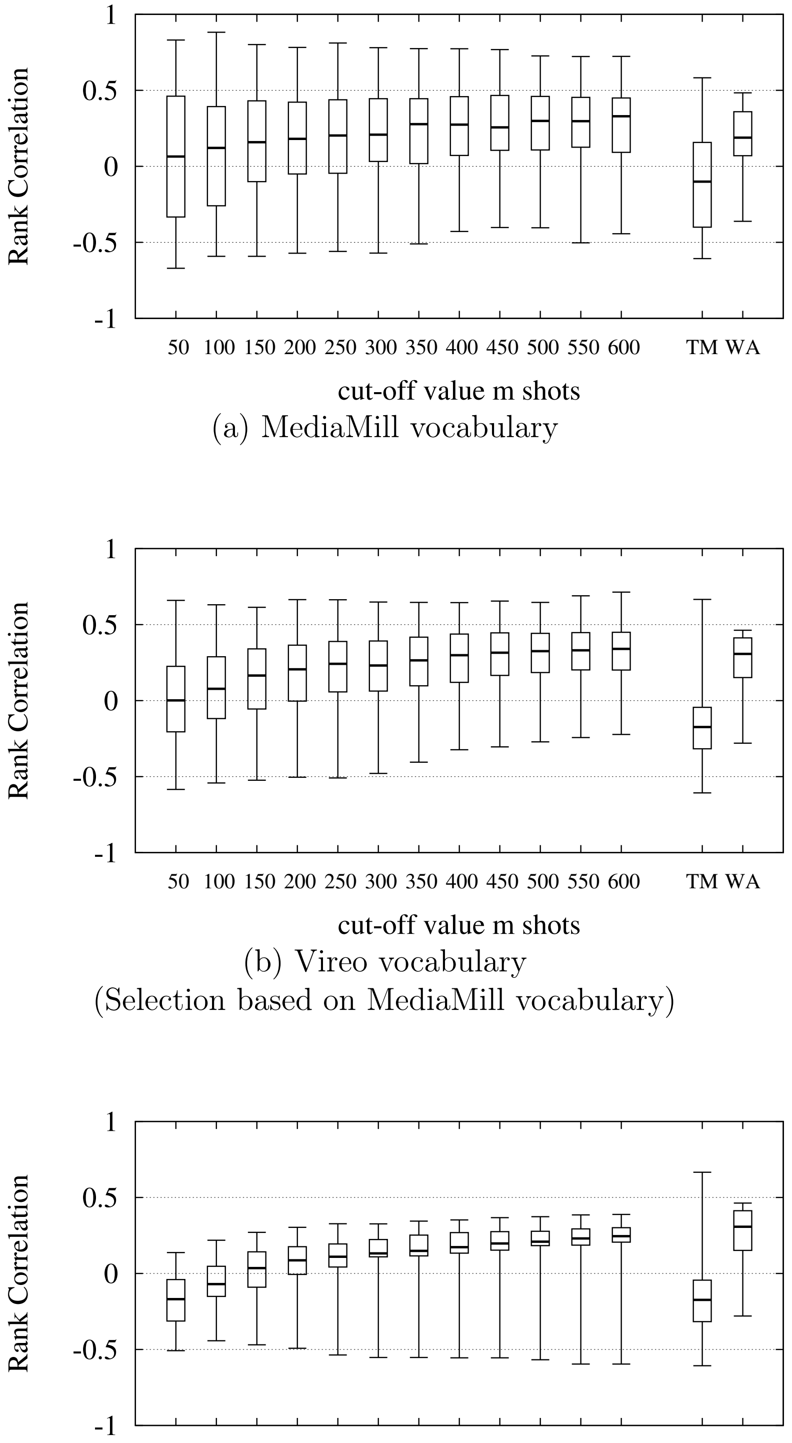
(b) Vireo vocabulary
(Selection based on MediaMill vocabulary)

(c) Vireo vocabulary
(Selection based on Vireo vocabulary)

Figure 4.1: *Evaluation of concept selections using the* average precision measure *(TM=text matching baseline, WA=wiki-article baseline).*

precision of the 24 queries using quartiles, see Section 4.4.1. Figure 4.1 (a) shows the results for the MediaMill vocabulary. The median average precision of the ADCS method rises until m=250. Afterwards, all three quartiles stay approximately the same. The lower outlier is always zero, meaning that there is always at least one concept selection which has an average precision of 0. From a cut-off value of m=200 the third quartile of the text matching baseline is approximately as high as the first quartile of the ADCS method. The wiki-article baseline performs better than the text matching baseline. From a cut-off value of m=250 fifty percent of concept selections of the ADCS method achieve a better performance than the wiki-article method (the third quartile of the WA is beneath the median of the ADCS method. Figure 4.1 (b) shows the results of the Vireo vocabulary where the initial text retrieval run was performed using the MediaMill concept descriptions. The results for the ADCS method are similar to the MediaMill vocabulary, only the upper outliers show a lower performance. For both baselines, text matching and wiki-article, 75% of the queries have a lower performance than 75% of the ADCS method. Figure 4.1 (c) shows the results of the Vireo vocabulary where the initial text retrieval run was performed using the Vireo concept descriptions themselves. Compared to Figure 4.1 (b) the baselines are unchanged since they do not depend on the initial retrieval run. The median concept selection also stays approximately the same. However, the distribution of the 24 queries is much denser around the median and the top performing queries (the upper outliers) also achieve a lower performance.

Rank Correlation Figure 4.2 shows the results of the *rank correlation* measure for the two concept vocabularies. The y-axis shows the *rank correlation* of the 24 queries using quartiles, see Section 4.4.1. Figure 4.2 (a) shows the results for the MediaMill vocabulary. Here, the first quartile and the median slowly increases with the cut-off value while the third quartile stays approximately the same at a rank correlation of 0.5. The upper outlier slowly decreases. With a high cut-off value of m=500, 75% of the concept selections from ADCS are better than 75% of the text matching method. The wiki-article method performs similarly to the ADCS method except that the upper outlier has a lower performance. Figure 4.2 (b) shows the results for the Vireo vocabulary where the initial text retrieval run was performed using the MediaMill concept descriptions. Here, the lower outlier, the first quartile and the median increase performance while the upper outlier remains approximately the same. For a cut-off value of m=600 nearly all concept selections of the ADCS method are better than the text matching baseline. For the wiki-article baseline the upper outliers are lower, otherwise the baseline performs similarly to the ADCS method. Figure 4.2 (c) shows the results for the Vireo vocabulary where the initial text retrieval run was performed using the Vireo concept descriptions themselves. Compared to Figure 4.2 (b) the baselines are unchanged since they do not depend on the initial retrieval run. The distribution of concept selections concentrates close to the upper



(a) MediaMill vocabulary

(b) Vireo vocabulary
(Selection based on MediaMill vocabulary)

(c) Vireo vocabulary
(Selection based on Vireo vocabulary)

Figure 4.2: *Evaluation of concept selections using the* rank correlation measure *(TM=text matching baseline, WA=wiki-article baseline).*

outlier, with an increasing cut-off value m. For all cut-off values it holds that 50% of the concept selections of the wiki-article baseline have a higher rank correlation than 75% of the concept selections from the ADCS method.

4.5 Summary and Discussion

This chapter introduced the Annotation-Driven Concept Selection (ADCS) method to select good concepts and set their weights using a textual query. The method is based on the construction of a textual representation of the development collection which was created to train concept detectors. The textual representation was built from the output of automatic speech recognition software and descriptions of the occurring concepts in a shot, which allows shots to be found for information needs looking for visual content. The concept descriptions consisted of the names, definitions and Wikipedia articles of concepts. For a new query, the following procedure is used to produce concept selections and weightings. First, the textual query is evaluated on the textual representation of the development collection and produces a ranking of shots. Second, the ADCS method estimates the occurrence probability of a concept in relevant shots, by using the score of the returned shots as weights. Here, higher ranked shots had a higher influence on the estimate and only the top-m shots were considered, which was referred to as the cut-off value. Finally, the estimated probability was used to calculate the Mutual Information of each concept with relevance. Then, the n concepts with the highest Mutual Information were selected to be used for retrieval in the actual search collection.

The evaluation of the ADCS method was performed on the TRECVid 2005 test collection using the *average precision* measure, see Hauff et al. (2007), and the *rank correlation* measure, see Huurnink et al. (2008). We investigated the following two concept vocabularies. First, the MediaMill vocabulary comprising 101 concepts. Second, the Vireo vocabulary comprising 374 concepts. For comparison, we used the baseline method text matching, proposed by Huurnink et al. (2008), and our own wiki-article method, proposed in previous work (Hauff et al., 2007).

The ADCS method performed stably using both concept vocabularies, both in terms of average precision and in rank correlation, with mildly increasing performance for higher cut-off values. For both measures, the performance was better for the MediaMill vocabulary than for the Vireo vocabulary. A possible reason is that in a larger concept vocabulary it is easier to select less useful concepts. Furthermore, the Vireo concept vocabulary performed better with a textual representation using the MediaMill vocabulary. Our interpretation of this is that a too verbose shot description (which results from a larger concept vocabulary) introduces less useful terms to the description. In terms of average precision, the ADCS method often returned better concept selections than the two baselines, Term Matching and wiki-articles. We propose that this originates from the more elaborate textual description of shots (compared to the ones of a single concept), which allows the text retrieval method to give higher ranking to shots in which multiple important concepts occur. This, in turn, results in a higher ranking of both concepts. In terms of rank correlation, the wiki-article method performed similarly to the ADCS method.

As there is no evaluation method for the accuracy of assigned *concept weights*, the evaluation of the produced weights are left to future work. Furthermore, we propose to investigate other shot description methods than Wikipedia articles.

The influence of the ADCS method on the search performance is evaluated in Chapter 5 and Chapter 6.

Chapter 5

Video Shot Retrieval

This chapter is based on Aly et al. (2008a) and Aly et al. (2009).

5.1 Introduction

This chapter presents the Probabilistic Ranking Framework for Unobservable Events (PRFUBE) for concept-based video shot retrieval. The PRFUBE is an application of the more general URR framework, presented in Chapter 3, for the document representations of binary concept occurrences. Compared to most other video shot retrieval functions described in Chapter 2 it has the following advantages:

(1) The PRFUBE takes into account all possible combinations of occurrence and absence of the considered concepts. Therefore, the PRFUBE explicitly models the case that concepts do not occur in relevant shots.

(2) The retrieval function of the PRFUBE is based on concept occurrences and is derived from the established Probability of Relevance Ranking Principle.

(3) The PRFUBE combines the retrieval function and the document representation uncertainty, originating from the detection of multiple concepts theoretically motivated by the URR framework.

The remainder of this chapter is structured as follows: In Section 5.2, the background concerning the ranking of binary document representations in text retrieval is given. Afterwards, Section 5.3 proposes the PRFUBE. Section 5.4 describes experiments which show the benefits of using the proposed framework. Finally, Section 5.5 ends this chapter with a summary and discussions.

5.2 Background: Ranking Binary Representations

In this section, the well-known Probability Ranking Principle for IR (PRP) (Robertson, 1977) is introduced because the ranking function of the PR-FUBE, proposed in this chapter, is derived from this principle. Afterwards, work from probabilistic indexing in text retrieval is described, because of the similarity to the proposed framework.

5.2.1 The Probability Ranking Principle for IR

The PRP is a general principle stating that it is optimal to rank by the posterior probability of relevance given an abstract document representation $\vec{f}$ of a document, $P(R|\vec{F} = \vec{f})$. In the literature, there are two ways for the calculation of the posterior probability. First, discriminative models, which map a document representation $\vec{f} \in dom(\vec{F})$ to a probability, see for example Yan (2006). Second, generative models which model the probability of relevance by the probability of the document representation $\vec{f}$ given relevance and the prior probability of the representation in the collection, see Bishop (2006) for a more detailed discussion. In this chapter we will use discriminative models, for which the PRP can be defined as:

$$retfunc_{PRP}\langle \vec{F}, w\rangle(\vec{f} : dom(\vec{F})) = \frac{w(\vec{f})P(R)}{P(\vec{F} = \vec{f})} \tag{5.1}$$

with

$$w(\vec{f}) = P(\vec{F} = \vec{f}|R)$$

Here, the probability of the document representation given relevance $P(\vec{F} = \vec{f}|R)$ is a query-specific weight and the prior probability of the document representation is a collection statistic, which can be calculated during indexing time. Furthermore, the relevance prior $P(R)$ can be ignored for the calculation of a ranking score value, since it is constant per query. Spärck-Jones et al. (2000); Fuhr (1992) give an overview of the vast amount of literature on this topic.

5.2.2 Probabilistic Indexing

An alternative to the assumption that librarians can assign fixed indexing terms to documents is probabilistic indexing, where a librarian, or a computer, only assigns probabilities as to whether a document would be indexed under a given index term. Croft (1981, 1983) integrates the probabilistic indexing approach with the binary independence model, by calculating the *expected* binary independence score from the binary independence model (BIM), see Equation 2.11, given the assigned probabilities. However, Fuhr (1989) shows that this retrieval function is not rank preserving to the PRP.

A reason for this was given in Section 3.3.5. Instead, Fuhr (1989) proposes another retrieval function:

$$retfunc_{ProbIdx}\langle d_m, w\rangle(P(t_i|d_m), \ldots P(t_n|d_m)) = \\ \prod_i^n \left[\frac{w(T_i)}{P(T_i)}P(T_i|d_m) + \frac{1-w(T_i)}{1-P(T_i)}P(\bar{T}_i|d_m)\right] \quad (5.2)$$

with

$$w(T_i) = P(T_i = 1|R)$$

Here, $w(T_i)$ is the probability that a relevant document is indexed with term T_i, $P(T_i)$ is the prior probability of a document being indexed with T_i and $P(T_i|d_m)$ is the probability that documents, of which we have knowledge d_m, are indexed with term T_i. The knowledge d_m is similar to the confidence scores for a video shot $\vec{o}$. The ranking function in Equation 5.2 is similar to the ranking framework proposed in this chapter and both frameworks are compared in Section 5.3.3.

5.3 Probabilistic Ranking Framework for Unobservable Binary Events

This section applies the UUR framework to the Probability of Relevance ranking function of uncertain binary concept occurrence representations, which results in the PRFUBE.

5.3.1 The Ranking Function

The URR framework requires a ranking function to rank documents under a known document representation. Because of the similarity between binary concept occurrences and binary index term assignments, the PRP is used. For a given document representation of concept occurrences $\vec{c}$, the retrieval function is derived from Equation 5.1:

$$retfunc_{PRPC}\langle \vec{C}, w\rangle(\vec{c} : dom(\vec{C})) = \frac{w(\vec{c})P(R)}{P(\vec{C} = \vec{c})} \quad (5.3)$$

with

$$w(\vec{c}) = P(\vec{C} = \vec{c})|R)$$

Here, PRPC is the probability of relevance ranking function based on a concept-based document representation $\vec{C}$. The query-specific weight $w(\vec{c})$ is the probability of a concept-based document representation given relevance. Furthermore, $P(\vec{C} = \vec{c})$ is the prior of this document representation and $P(R)$ is the relevance prior, which can be left out since it does not influence the ranking.

5.3.2 Framework Integration

However, the concept-based document representation is uncertain. As a result, the score of a document is also uncertain. Therefore, let $score_q := new\ retfunc_{PRPC}\langle\vec{C}, w\rangle$ be the score function for the current information need based on a concept document representation $\vec{C}$, which is derived from the ranking function Equation 5.3. For a document d the document specific random variable for its representation is $\vec{C}(d)$[1]. Furthermore, let $S(d) = score_q(\vec{C}(d))$ be the random variable "the score for document d". We now define the two components of the URR framework, the expected score and its variance.

Expected Score The expected score for the probability of relevance ranking function from Equation 5.3 is defined as:

$$E[S(d)|\vec{o}] = \sum_{\vec{c} \in dom(\vec{C}(d))} score_q(\vec{c}) P_\Omega(\vec{C}(d) = \vec{c}|\vec{o}) \tag{5.4}$$

Here, $\vec{c}$ is one of $|dom(\vec{C}(d))| = 2^n$ possible representations of the n considered concepts. Each document representation $\vec{c}$ has the score $score_q(\vec{c})$. The expected score is the weighted average of these scores, according to the occurrence probability of the representation.

Variance of Score The second component of the URR framework is the variance of the score. Following the derivation in Equation 3.6, the variance of the score is calculated via the expected squared score:

$$\begin{aligned} \mathrm{var}[S(d)|\vec{o}] &= E[S(d)^2|\vec{o}] - E[S(d)|\vec{o}]^2 \text{ with} \\ E[S(d)^2|\vec{o}] &= \sum_{\vec{c} \in dom(\vec{C})} score_q(\vec{c})^2 P_\Omega(\vec{C}(d) = \vec{c}|\vec{o}) \end{aligned} \tag{5.5}$$

The variance expresses how much the possible scores vary for a given distribution of possible representations.

Combining the Components According to Section 3.3.4 documents should be ranked by a combination of the expected score and its standard deviation:

$$RSV(d) = \underbrace{E[S(d)|\vec{o}]}_{\text{Eq. 5.4}} - b \underbrace{\sqrt{\mathrm{var}[S(d)|\vec{o}]}}_{\text{Eq. 5.5}} \tag{5.6}$$

Here, b is the risk parameter representing the risk attitude of the retrieval engine. If $b > 0$, the retrieval engine is called risk-averse. For $b = 0$ the retrieval engine is risk neutral and for $b < 0$, we call the retrieval engine risk-loving. Equation 5.6 is the direct application of the URR framework on

[1] For an explanation of document specific document variables see Appendix A.

representations of binary concept occurrences. However, Equation 5.6 has a run-time complexity of $O(2^n)$ (that is because Equation 5.4 and Equation 5.5 have a run time complexity of $O(2^n)$, and is therefore only usable for small numbers of concepts, n.

5.3.3 PRFUBE: Operational Ranking Function

Since the retrieval function from Equation 5.6 is not applicable to realistic numbers of concepts, the ranking function has to be adjusted in order to be executed efficiently. The result will be the ranking framework proposed by this chapter. The following independence assumptions are made to make the computation more efficient:

$$P(\vec{C}|R) = \prod_{i}^{n} P(C_i|R) \tag{5.7}$$

$$P(\vec{C}) = \prod_{i}^{n} P(C_i) \tag{5.8}$$

Here, Equation 5.7 assumes conditional independence of all random variables C_i, given relevance; which is a common assumption in text retrieval. Equation 5.8 assumes that the concept variables are independent. These assumptions are also made by Fuhr (1989) for index term assignments.

Furthermore, recall that we make the following independence assumptions concerning the probability measure of concept detectors, described in Section 2.3.4:

$$P_\Omega(\vec{C}|\vec{o}) = \prod_{i}^{n} P_\Omega(C_i|o_i) \tag{5.9}$$

Expected Score By using the above independence assumptions and using the generative version of the probability of relevance ranking function from Equation 5.3, the expected score from Equation 5.4 can now be expressed as follows:

$$E[S(d)|\vec{o}] = P(R) \sum_{\vec{c} \in dom(\vec{C})} \prod_{i}^{n} \frac{P(C_i = c_i|R)}{P(C_i = c_i)} P_\Omega(C_i(d) = c_i|o_i) \tag{5.10}$$

Here, the query-specific constant $P(R)$ can be ignored. Additionally, because $\vec{C}$ is a vector of binary random variables the generalized distributive law can be applied, see Aji and McEliece (2000), which results in the operational calculation of the expected score:

$$E[S(d)|\vec{o}] =$$

$$\prod_{i=1}^{n} \Big[\underbrace{\frac{w(C_i)}{P(C_i)} P_\Omega(C_i(d)|o_i)}_{C_i \text{ occurs}} + \underbrace{\frac{1 - w(C_i)}{P(\bar{C}_i)} P_\Omega(\bar{C}_i(d)|o_i)}_{C_i \text{ is absent}} \Big] \tag{5.11}$$

with

$$w(C_i) = P(C_i|R)$$

Here, $w(C_i)$ is the occurrence probability of a concept given relevance. $P(C_i)$ is the concept prior, which is calculated according to Equation 3.11, and $P_\Omega(C_i|o_i)$ is the probability measure of a concept occurring in Ω given the confidence score o_i.

Variance of the Score By following the calculation in Equation 5.5, the variance of the score can be calculated through the expected squared score $E[S(d)^2|\vec{o}]$ and a similar derivation to the one of the expected score in Equation 5.3.3 results in:

$$E[S(d)^2|\vec{o}] = \prod_{i=1}^{n} \left[\left[\frac{w(C_i)}{P(C_i)}\right]^2 P_\Omega(C_i(d)|o_i) + \left[\frac{1-w(C_i)}{P(\bar{C}_i)}\right]^2 P_\Omega(\bar{C}_i(d)|o_i) \right] \quad (5.12)$$

from which we can calculate the variance:

$$\mathrm{var}[S(d)|\vec{o}] = E[S(d)^2|\vec{o}] - E[S(d)|\vec{o}]^2 \quad (5.13)$$

Combining the Components Using the expected score and its variance, the ranking score value of a document can be calculated as follows:

$$RSV(d) = \underbrace{E[S(d)|\vec{o}]}_{\text{Eq. 5.11}} - b \underbrace{\sqrt{\mathrm{var}[S(d)|\vec{o}]}}_{\text{Eq. 5.13}} \quad (5.14)$$

Here, the risk parameter b determines the mixture of the expected score and the variance and Equation 5.14 is the direct derivation the URR framework. However, we will show in an experiment in Section 5.4.5 that for a risk-loving setting ($b < 0$) the influence of the variance does not affect the search performance while with a risk-averse setting ($b > 0$) the search performance quickly degrades. Therefore, we always set $b = 0$ and only use the expected score in Equation 5.11 as the operational ranking function for binary document representations, PRFUBE. Its time complexity is in $O(n)$ with small constants (6 multiplications and 1 addition per concept). Therefore, it is usable for all common numbers of selected concepts.

Similarity to Fuhr's Retrieval Function Equation 5.11 is mathematically equivalent to the ranking function for probabilistic indexing from Fuhr (1989), see Equation 5.2. However, the difference lies in the considered probabilistic event space of the two methods: Fuhr (1989) (p. 59) considers the cross product between *all* queries, documents and possible index assignments to be events. A document is ranked by the probability of relevance given the knowledge a retrieval engine has of this document, by marginalizing over the uncertain index term assignments. Therefore, each document has exactly one score. On the other hand, the PRFUBE considers the original event space of the probability of relevance ranking principle, where the events are pairs of the current information need and the documents in the collection. With known concept occurrences, the documents are ranked under the probability of relevance given their *correct* concept occurrences, which correspond to the index term assignments. Since the concept occurrences are uncertain, documents have multiple possible scores and they are ranked by the expected score. As a result, while mathematically equivalent, the difference between Fuhr's ranking function in Equation 5.2 and the PRFUBE is the *interpretation* of the ranking process.

Similarity to the PMIWS If the part of Equation 5.11 marked with C_i *is absent* is left out, we have a ranking function which produces the same rankings as the Pointwise Mutual Information Weighting Scheme, PMIWS, from Zheng et al. (2006), see Equation 2.7. This suggests that Zheng's ranking function only considers the case where all concepts occur. In other words, Zheng's ranking function is the probability of relevance given the representation where all n concepts occur, multiplied by the probability that this representation was the correct one. Therefore, the main difference between the PRFUBE and the PMIWS ranking function is that the PRFUBE also considers cases where concepts can be absent in relevant shots.

5.3.4 Implementation

Algorithm 5.1 shows a pseudo code implementation for the operational PRFUBE described in Equation 5.11. First, in the procedure *retrievalrun*, concepts are selected using the Annotation-Driven Concept Selection, see Chapter 4. The PRFUBE appears as an instance of the traditional information retrieval process, as described in Section 1.2, since the treatment of multiple representations is not done by iterating over all possible representations. Instead, the ranking function iterates over all selected concepts – which results in the same scores as the iteration over all 2^n possible representations because of the distributive law (Aji and McEliece, 2000).

5.4 Experiments

This section presents the experiments which were performed to assess the performance of the proposed PRFUBE.

5.4.1 Experimental Setup

Collections This section describes the experimental setup. Statistics about the collections and concept detectors, which are used in the following exper-

Algorithm 5.1: Implementation of the video shots retrieval engine PRFUBE.

```
**Data**: Collection $\mathcal{D}$,
Query Features $\vec{QF}$,
Concept Statistic $P(C)$
w is the query-specific weight: $w(C_i) = P(C_i|R)$

$retfunc_{PRFUBE}\langle\vec{O}, w\rangle(\vec{o} : dom(\vec{O})) =$
        $\prod_i^n \left[\frac{w(C_i)}{P(C_i)} P(C_i|o_i) + \frac{1-w(C_i)}{P(\bar{C}_i)} P(C_i|o_i)\right]$

$retrievalrun(\vec{qf} \in dom(\vec{QF})) :$
begin
    // Score Function Definition
    $(\vec{C}, w) := selectNweight_{ADCS}(\vec{qf})$ // from Chapter 4
    // confidence scores corresponding to selected concepts
    $\vec{O} := (O_{c_1}, \ldots, O_{c_n})$
    $score_q := new\ retfunc_{PRFUBE}\langle\vec{O}, w\rangle$
    // Match and Combine
    foreach Document d in $\mathcal{D}$ do
        Append $(d, score_q(\vec{o}(d)))$ to ranking
    end
    return $sort(ranking, ranking.score\ \ DESC)$
end
```

iments are summarized in Table 5.1. We consider two classes of collections. First, the primary collections, TRECVid 2005 (tv05t) and TRECVid 2007 (tv07t), which will be used in graphs to visualize the results of the experiments. Second, secondary collections which will only be used in summaries. The primary collections were chosen since they are from two different video domains and they are often used in the literature.

Retrieval Functions We compare the PRFUBE against a set of retrieval functions representing the uncertainty classes presented in Section 2.4. Table 5.2 shows and overview of the used retrieval functions. Note, it would have been interesting to compare PRFUBE with the "Probabilistic Model for combining diverse Knowledge Sources in Multimedia" (PKSrc) by Yan (2006). However, this ranking function was not included since it requires confidence scores on a development collection which were only available for the TRECVid 2005 collection. For display purposes, the Pointwise Mutual Information Weighting Scheme PMIWS and the CombMNZ method are used for detailed experiments. The remaining retrieval functions are only included in summaries.

Collection	Shots	Domain	Queries	Detectors	Concepts	Training Data
Primary Collections (used in all experiments)						
tv05t	45,765	News	24	MM101	101	tv05d
tv07t	18,142	G.TV	24	Vireo	374	tv05d
Secondary Collections (used only in summaries)						
tv06t	79,484	News	24	Vireo	374	tv05d
tv08t	35,766	G.TV	48	Vireo	374	tv05d
tv08t	35,766	G.TV	48	MM09	64	tv07d
tv09t	61,384	G.TV	24	MM09	64	tv07d

Table 5.1: *Statistics over the used collections in the presented experiments. The following abbreviations are used: tvXXt: Search collection of year 20XX, News: Broadcast News, G.TV: General Dutch Television. The detector sets are described in the following publications: MM101 (Snoek et al., 2006), Vireo (Jiang et al., 2010), MM09 (Snoek et al., 2008).*

Ret. Func.	Description	Definition
CombMNZ*	Multiply non-zero (see Sec. 2.4.2)	$\prod_i P(C_i\|o_i)$
CombSUM	Unweighted sum of scores (see Sec. 2.4.2)	$\sum_i P(C_i\|o_i)$
PMIWS*	Pointwise Mutual Information Weighting Scheme (see Sec. 2.4.2)	$\sum_i \log(\frac{P(C_i\|R)}{P(C_i)})P(C_i\|o_i)$
Borda-Count	Rank Based (see Sec. 2.4.3)	$\sum_i \mathit{rank}(P(C_i\|o_i)$
BIM	Binary Independence Model (see Sec. 2.4.4)	$\sum_i c_i' \log(\frac{p(1-q)}{q(1-p)})$
ELM	Expected Concept Occurrence Language Model ($\lambda = 0.1$) (see Sec. 2.4.5)	$\prod_i \left[\lambda P(C_i\|o_i) + (1-\lambda)P(C_i\|\mathcal{D})\right]$

Table 5.2: *Retrieval functions (Ret. Func.) used in the experiments. Retrieval functions marked by * are used in detailed comparison while others are only used in summaries.*

Concept Selection and Weighting For the concept selection and weighting two baselines are investigated, which are discussed in Section 5.4.2. Furthermore, for automatic concept selection and weighting, the Annotation-Driven Concept Selection (ADCS) method was used, see Chapter 4. Two different development collections were used for the estimation of the query-specific weights $P(C|R)$. First, for the MM101 and Vireo detector sets, descriptions of the TRECVid 2005 development collection based on the MM101 vocabulary are used. Second, for the estimation of the weights for the MM09 detectors, descriptions from the TRECVid 2007 development collections based on the 64 concepts are used. The shot description consisted of the automatic speech recognition output plus the concept definitions and the corresponding Wikipedia articles of the occurring concepts. For the initial text retrieval run the general purpose retrieval engine PF/Tijah (Hiemstra et al., 2006) was used to rank shots in the development collection.

5.4.2 Baseline and User Vote Concept Selection

Baseline In order to compare the search performance of the ADCS method we use our previously propose approach, see Hauff et al. (2007), as a baseline (called wiki-articles). The approach represents each concept by a Wikipedia article and executes a query first on this collection to attain scores for each concept. The score is then linearly transformed into the interval within [0 : 1] to be able to interpret them as the occurrence probability of a concept given relevance, $P(C|R)$, which is used in by the PRFUBE and PMIWS method.

$$P(C|R) = \frac{score - min}{max - min} range + lowest \tag{5.15}$$

Here, *score* is the score from the text retrieval engine, *min* and *max* are the minimum and maximum returned score. Furthermore, *range* is the width of the interval and *lowest* is the lowest value of the interval. For the TRECVid 2005 collection, the empirical optimal parameter setting is *range*=0.60 and *lowest*=0.05. For this collection, the baseline achieves a performance of 0.051 MAP. Furthermore, for TRECVid 2007 the parameter setting was *range*=0.20 and *lowest*=0.05 and the proposed baseline achieved 0.013 MAP.

Golden Standard In order to create a golden standard[2], we investigate how effective humans can select concepts. Here, the results from a user study with 23 users are investigated. The users were asked to select all concepts, which they thought were related to an information need. Afterwards, the occurrence probability of a concept given relevance, $P(C|R)$, was estimated by the relative number of users who selected the concept C:

$$P(C|R) = \frac{numUser(C)}{numUser} \tag{5.16}$$

[2] A performance level which indicates a good performance of an automatic concept selection method.

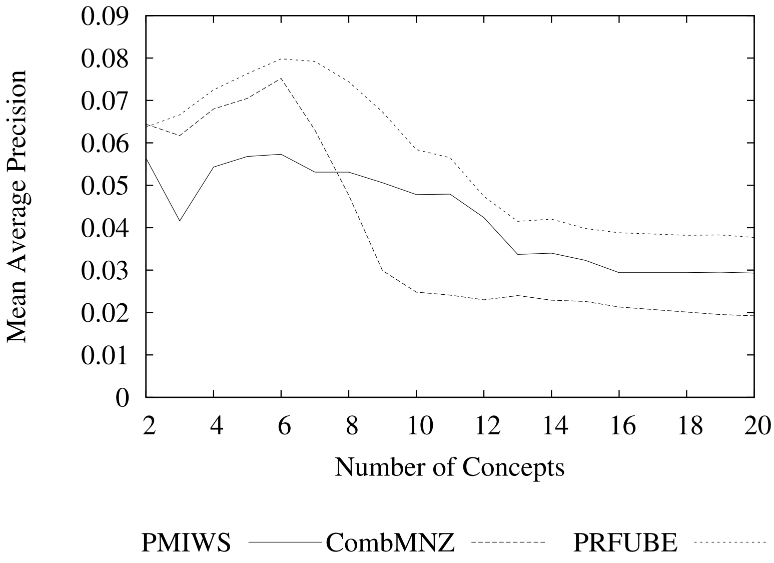
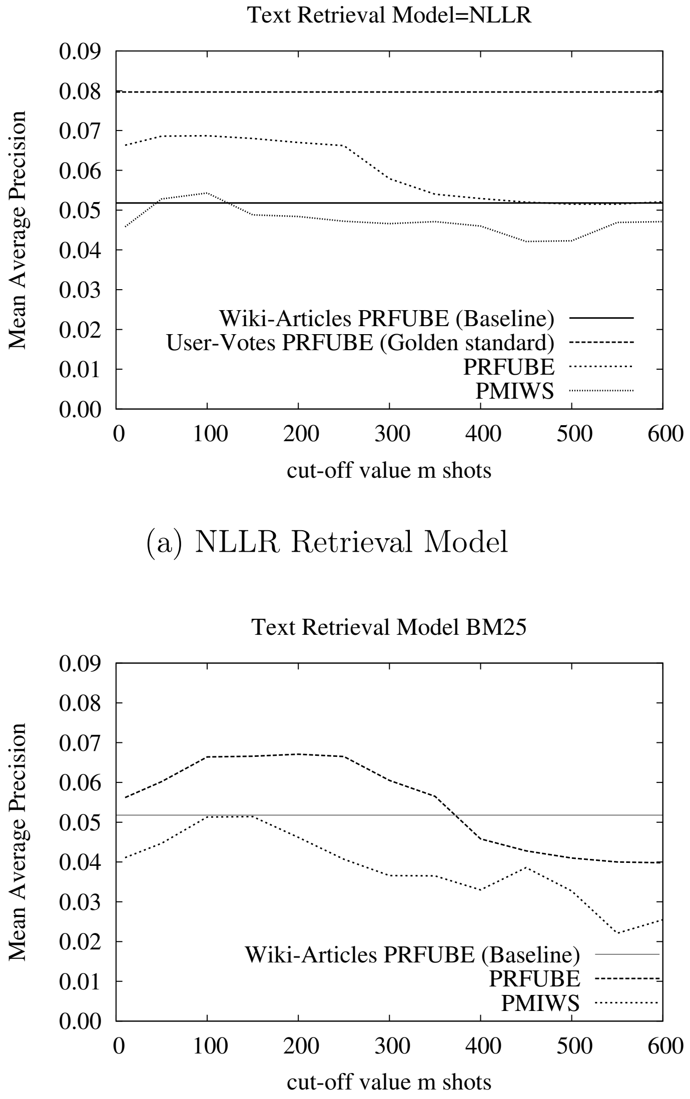


Figure 5.1: *Concept selection and parameter estimation from user-study (called Golden Standard).*

Here, $numUser(C)$ is the number of users who selected concept C and $numUser$ is the total number of users having participated in the study. The underlying assumption is that more users will select a concept if it occurs more often in relevant shots. Due to constraints in resources and the large number of concepts the study was limited to the TRECVid 2005 topics and the MM101 vocabulary. Note, since multiple users are involved in the search process, the practical application of such a concept selection method is only applicable in collaborative search scenarios, as for example demonstrated by Adcock (2007).

Figure 5.1 shows the results of the combination of the concepts selected by the users form the study. Only 20 concepts out of the 101 concept in the MM101 vocabulary were plotted, because this was the maximum number of selected concepts for an information need. All methods show their best performance at approximately six concepts. However, from the seventh concept onwards the performance decreases for all methods. Investigations revealed that the eighth concept, *Crowd*, for the query 0156 *Tennis player on the court* which had the highest average precision did appear less often in relevant than in non-relevant shots. The reason is that the concept referred to a courtroom while the users assumed it to refer to a tennis court. PRFUBE performs in all cases better than the other methods. The maximum performance of 0.078 MAP from the PRFUBE is used as a golden standard to judge the retrieval performance with the ADCS method in Section 5.4.3.



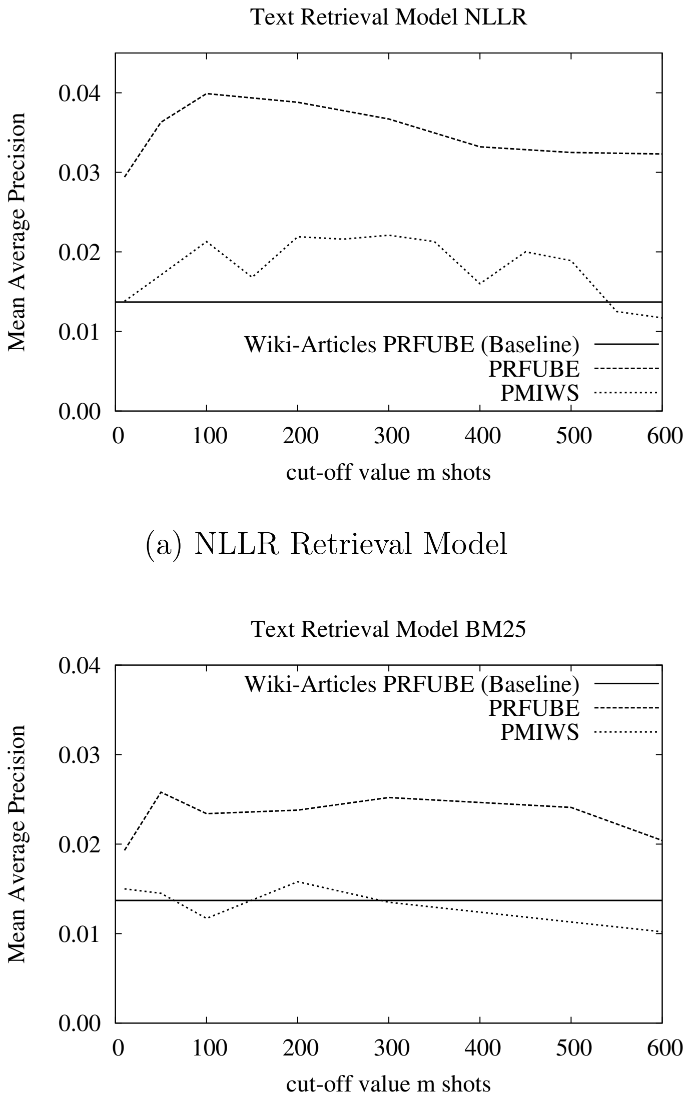
(a) NLLR Retrieval Model

(b) BM25 Retrieval Model

Figure 5.2: *Performance of retrieval runs on the TRECVid 2005 test using the Annotation-Driven Concept Selection method, see Chapter 4, varying the cut-off value m and the text retrieval model.*

5.4.3 Annotation-Driven Concept Selection

This section describes experiments conducted to investigate the performance that can be achieved with the PRFUBE together with the automatic ADCS method. First, the results for the two primary collections, TRECVid 2005 and TRECVid 2007, are investigated. Then, a summary of the search performance of all used collections is given.



(a) NLLR Retrieval Model

(b) BM25 Retrieval Model

Figure 5.3: *Performance of retrieval runs on the TRECVid 2007 test using the Annotation-Driven Concept Selection method, see Chapter 4, varying the cut-off value m and the text retrieval model.*

TRECVid 2005 Figure 5.2 (a) shows results of the PRFUBE and the PMIWS method for the TRECVid 2005 search task using the NLLR text retrieval model. The chosen number of concepts was experimentally set to produce the optimum performance which was ten concepts for the PRFUBE model and seven for the PMIWS method. The x-axis shows the cut-off value m used for the ADCS method, see Section 4.3.2. The y-axis depicts the mean average precision. PRFUBE performs best and achieves 0.069 MAP at a cut-off value of m = 150 shots. Afterwards, the performance declines and stabilizes at around 0.055 MAP. The PMIWS method achieves 0.054 MAP at a cut-off value of $m = 100$ shots. Afterwards, it stabilizes at around 0.046. Recall, the wiki-articles baseline achieves a 0.051 MAP and the golden standard from the user study achieves 0.079 MAP, see Section 5.4.2. Since both methods do not depend on the cut-off value m, they are drawn as constant lines in the graph. The improvement of the estimations for PRFUBE at a cut-off value of $m = 150$ shots against the wiki-articles baseline and the PMIWS retrieval function was tested for significance using a two-sided, paired Wilcoxon signed rank test with a significance level of 0.05.

In Figure 5.2 (b) the experiment is repeated using the BM25 text retrieval model by Robertson and Walker (1994). For cut-off values of $m < 200$ shots the graphs look similar. However, with increasing m the MAP of both ranking models, PRFUBE and PMIWS, decreases further than with the NLLR retrieval model.

TRECVid 2007 Figure 5.3 (a) shows the results of the PRFUBE and the PMIWS runs for the TRECVid 2007 collection. Note, compared to our prior publication (Aly et al., 2009), on which this chapter is based, this experiment uses newer articles from Wikipedia, which improved the performance of the PRFUBE. Furthermore, the range of considered cut-off values is limited to the interval [0 : 600] to make it comparable with the TRECVid 2005 experiment in Figure 5.2. The number of concepts was chosen in the same way as for the TRECVid 2005 collection. Here, the optimum was 45 concepts for the PRFUBE and 15 for the PMIWS model. The axes depict the same as in Figure 5.2. The PRFUBE shows the best performance at a cut-off value up to $m = 100$ shots, resulting in 0.039 MAP search performance. Afterwards, the performance decreases to 0.030 MAP and stabilizes. The PMIWS model performs less stably. The best search performance of 0.021 MAP is achieved at a cut-off value of $m = 300$ shots. The wiki-articles baseline only achieves 0.013. The improvement of PRFUBE at a cut-off value of $m = 100$ shots, compared to the wiki-articles baseline was significant according to a two-sided, paired Wilcoxon signed rank test with a significance level of 0.05

Figure 5.3 (b) shows the performance of the PRFUBE and PMIWS ranking model using with the BM25 text retrieval model. The performance of the PRFUBE retrieval model decreases from 0.025 MAP to 0.020 MAP. The PMIWS method has the highest performance at a cut-off value of $m = 200$ shots. Afterwards the performance decreases to 0.010 MAP.

Summary Table 5.3 summarizes the retrieval performance of the seven considered retrieval functions over five collections. For the TRECVid 2008 collection the search performance for two detector sets is evaluated. For each retrieval function, the table reports four numbers. First, the optimal performance in MAP that the method was able to achieve. Second, the rank of this performance within the seven functions is shown in brackets (in the case of ties, we assign all tied systems the higher rank). Third and fourth,

Collection Ret. Func.	tv05t MM101	tv06t Vireo	tv07t Vireo	tv08t Vireo	tv08t MM09	tv09t MM09	Avg. Rank
CombMNZ	0.064 (3) 10/8	0.033 (4)† 700/30	0.028 (3) 100/20	0.024 (6)† 10/15	0.042 (7)† 100/30	0.045 (6)† 100/10	4.7
CombSUM	0.047 (6) 100/1	0.025 (5)† 100/2	0.028 (3) 100/8	0.017 (7) 100/4	0.036 (6)† 200/4	0.041 (7)† 100/4	5.7
PMIWS	0.054 (4) 100/8	0.039 (3) 200/30	0.021 (5) 200/15	**0.041** (1) 50/4	**0.058** (1) 50/4	0.067 (2) 50/2	2.7
Borda-Count	0.050 (5)† 10/15	0.012 (7)† 100/10	0.020 (6)† 50/20	0.030 (5) 10/15	0.045 (5)† 10/2	0.058 (5) 10/8	5.5
BIM	0.044 (7)† 10/8	0.024 (6)† 100/2	0.026 (4) 100/8	0.037 (4) 100/4	0.050 (4) 50/2	0.063 (4) 50/2	4.8
ELM	**0.071** (1) 10/8	0.040 (2) 600/30	0.031 (2) 50/10	0.040 (3) 100/4	0.050 (3) 10/2	0.064 (3) 50/2	2.3
PRFUBE	0.069 (2) 150/10	**0.043** (1) 600/30	**0.039** (1) 100/45	0.041 (2) 100/4	0.056 (2) 100/4	**0.068** (1) 50/2	1.5

Table 5.3: *Mean average precision of the retrieval functions described in Table 5.2. For each retrieval function, the first row indicates the search performance and its rank within the seven retrieval functions and the second row states the cut-off m value and the number of concepts n which were used to achieve this performance. The † symbol indicates that the method is significantly* worse *than the best method for this collection (Rank (1)), according to a two-sided, paired Wilcoxon signed rank test with a significance level of* 0.05.

the cut-off value m and the number of concept n with which this performance was achieved. On the right, the average rank of the method over the six runs is reported.

The PRFUBE is on average the best retrieval function. In three of the six cases it the second best method. However, in all three cases the differences is not significant according to a two-sided, paired Wilcoxon signed rank test with a significance level of 0.05.

5.4.4 Retrospective Experiments

In this section, we report retrospective experiments conducted to reveal what performance is achievable, given a perfect weight estimation for the occurrence probability of a concept given relevance, $P(C|R)$. The perfect estimation is calculated, under the assumption that we know about the relevance of all shots. Similar to the estimation of the concept prior in Section 3.3.5, the expected occurrence probability of a concept given relevance is calculated as follows.

$$P(C|R) = E[P(C|R)|o(d_1), ..., o(d_N)] = \frac{\sum_{d \in \mathcal{D}, r(d)=1} P_\Omega(C|o(d)}{|\{d \in \mathcal{D}|r(d) = 1\}|}$$

Here, the expected number of relevant shots in which the concept occurs is calculated and is divided by the total number of relevant shots. Note, for many detectors this estimate was much smaller than an intuitive value. For example, for the TRECVid 2005 query "0150 Find shots of Iyad Allawi", this estimate resulted for the concept *Allawi* in only 0.007, whereas it should be 1, by the definition of the concept. For the TRECVid 2005 collection and the MM101 vocabulary we found that, on average, only 40% of the top-10 concepts selected by users from the user-study, see Section 5.4.2, were found under the concepts selected by the retrospective method.

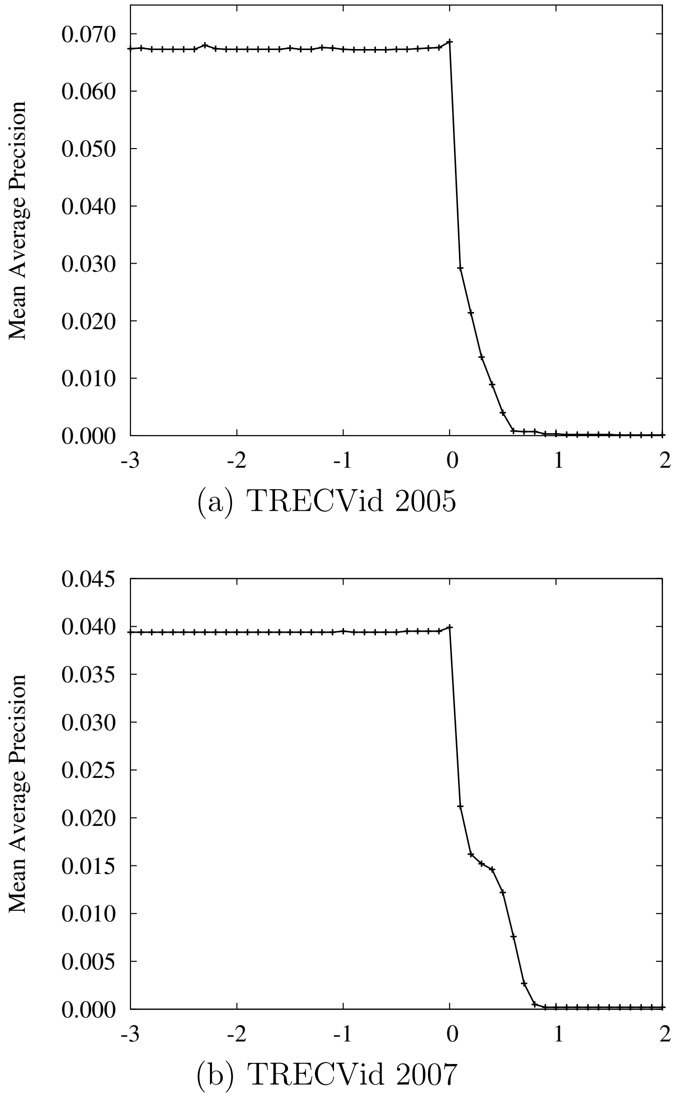
Table 5.4 shows the summary of the experimental results of the experiments using perfect parameter estimation. The setup of the table is similar to the one summarizing the results of the automatic concept selection task. Except in the TRECVid 2008 collection with the MM09 detector set the PRFUBE always shows the best performance.

5.4.5 Risk Parameter Study

Figure 5.4 shows the influence of the risk parameter b on the detector performance of the TRECVid 2005 and TRECVid 2007 collection. For both experiments, the outcome is similar. For a risk-averse attitude ($b > 0$) the search performance quickly decreases to virtually zero and for a risk-loving attitude the search performance stays approximately the same. Therefore, using a risk neutral setting for PRFUBE is optimal.

Collection	tv05t	tv06t	tv07t	tv08t	tv08t	tv09t	Avg. Rank
Ret. Func.	MM101	Vireo	Vireo	Vireo	MM09	MM09	
CombMNZ	0.070 (5)† 2	0.032 (6) 5	0.018 (3) 18	0.015 (6) 3	0.028 (7)† 2	0.067 (7)† 4	5.7
CombSUM	0.062 (6)† 5	0.033 (5) 5	0.018 (3) 5	0.017 (5) 3	0.035 (5)† 5	0.110 (4)† 2	4.7
PMIWS	0.077 (2) 8	0.039 (2) 5	0.025 (2) 12	0.030 (3) 12	**0.070** (1) 5	0.138 (2) 4	2.0
Borda-Count	0.074 (4) 5	0.036 (3)† 1	0.010 (5)† 3	0.017 (5) 1	0.040 (4) 5	0.096 (5)† 5	4.3
BIM	0.054 (7) 8	0.035 (4) 35	0.025 (2)† 5	0.034 (2) 35	0.050 (3)† 8	0.095 (6)† 8	4.0
ELM	0.076 (3) 5	0.039 (2) 5	0.017 (4) 3	0.025 (4) 3	0.032 (6) 5	0.112 (3) 5	3.7
PRFUBE	**0.083** (1) 12	**0.043** (1) 5	**0.034** (1) 20	**0.036** (1) 35	0.067 (2) 5	**0.140** (1) 4	1.2

Table 5.4: *Mean average precision performance of retrospective experiments, setting the query-specific parameter $P(C|R)$ with knowledge of relevance. For each retrieval function, the first row indicates the search performance and its rank within the seven retrieval functions. The second row, states the number of concepts n which were used to achieve this performance. The † symbol indicates that the method is significantly* worse *than the best method for this collection (Rank (1)), according to a Wilcoxon signed-rank test with a significance level of* 0.05.



(a) TRECVid 2005

(b) TRECVid 2007

Figure 5.4: *Risk parameter b for the ranking function $RSV(d) = E[S(d)|\vec{o}] - b\,\sqrt{var[S(d)|\vec{o}]}$.*

5.5 Summary and Discussion

Theoretical Contribution This chapter proposed the Probabilistic Ranking Framework for Unobservable Events (PRFUBE), where binary events refer to concept occurrences. The PRFUBE was derived from the Uncertain Representation Ranking (URR) framework, proposed in Chapter 3. The URR framework requires a ranking function for the case that the document representation was known. Here, the probability of relevance ranking function was chosen (Robertson, 1977). First, a direct derivation of the URR framework for binary concept representations was given. However, the resulting ranking function had an exponential run-time complexity in the number of considered concepts. Therefore, an operational ranking function, PRFUBE, was proposed, which had a linear run-time complexity. The PRFUBE explicitly models the case that a considered concept could be absent from relevant shots. Besides, the selection of concepts it requires the concept prior probability, a collection statistic, and the occurrence probability of a concept given relevance, as a query dependent weight. For concept priors the estimation method proposed in Section 3.3.5 was used and several methods for the concept selection and weighting were investigated in the experiments which will be discussed below. It was found that the influence of the variance did not affect the search performance for a risk-loving attitude and quickly decreases the search performance for a risk-averse attitude. Therefore, a risk neutral attitude was used in all experiments – which does not take the variance into account.

Retrieval Performance In the experiments, the benefits of the PRFUBE was evaluated together with three estimation methods for the query-specific parameter, the occurrence probability of a concept given relevance.

(1) A baseline from our own previous work (Hauff et al., 2007) (called wiki-articles) was used for which the parameter was a linearly scaled version of a text retrieval score for a Wikipedia article describing the concept.

(2) An estimation through the results of a user study, in which users selected important concepts, was used as a golden standard.

(3) The application of the Annotation-Driven Concept Selection (ADCS) method, proposed in Chapter 4.

The experiments were performed on five different collections from the TRECVid years 2005-2009. The TRECVid 2005 and TRECVid 2007 collection were selected as primary collections, on which the experiments were carried out in higher detail. The PRFUBE method was compared to a set of six retrieval functions from the literature.

The wiki-articles baseline showed unstable performance and there was no guidance to set the scaling parameters, required by this method. For the TRECVid 2005 collection the performance was 0.050 MAP and for the TRECVid 2007 collection 0.010 MAP was achieved. The user selection from the user-study showed good results, resulting for TRECVid 2005 in a search performance of 0.078 MAP for PRFUBE. The ADCS method automatically set the parameters. It was shown that the search performance was stable for many values of the cut-off parameter used in the ADCS method. The same holds for a variation of the text retrieval model of the initial text search in the ADCS method. In both collections the PRFUBE was able to improve upon the baseline, with 0.069 MAP for the TRECVid 2005 and 0.036 MAP for the TRECVid 2007 collection. These improvements were shown to be significant according to a Wilcoxon signed-rank test with a significance level of 0.05.. Two other retrieval functions, CombMNZ and PMIWS (see above), also improved with the parameter estimation. In the overall comparison, PRFUBE was on average the best retrieval system, with an average rank of 1.5 among all seven examined retrieval functions in all six collection-detector set combinations. Furthermore, the three times where PRFUBE was the second best system, it was not significantly worse than the best system.

Retrospective Experiments In order to show that the search performance of PRFUBE further improves with better parameter settings, we performed retrospective experiments, where the parameters were calculated under the knowledge of the relevance of shots by the average of the concept occurrence probability given relevance. In a comparison of all retrieval functions on all collection-detector set combinations the PRFUBE was only worse than the Expected Concept Occurrence Language Model once.

It was found that only 40% of the top-10 concepts for the queries from the TRECVid 2005 query set selected by the user study, see Section 5.4.2 were also in the top-10 concept selected by the retrospective experiments. A likely explanation, for why the calculated parameters improved the performance is that the ranking function of PRFUBE does not consider the *performance* of concept detectors. The investigation of concept detector performance is a current research topic (Yang and Hauptmann, 2008a; Wei et al., 2008). Therefore, the integration of the results of this research into the PRFUBE is proposed for future work.

Chapter 6

Video Segment Retrieval

This chapter is based on Aly et al. (2010).

6.1 Introduction

Video retrieval engines have usually concentrated on retrieval at the shot level, with a shot being a visually distinct group of images (Smeaton et al., 2009). Not as much attention has been paid to searching for longer video segments. This chapter proposes that users may relate better to these longer video segments. However, current video retrieval models are difficult to adapt to these segments since they are tailored to find shots which are most often represented by a single key frame. This chapter proposes an application of the language modeling retrieval function for video segment retrieval based on unknown concept occurrences applying the general Uncertain Representation Ranking (URR) framework, proposed in Chapter 3. This application is termed: Uncertain Concept Occurrence Language Model (UCLM) framework. To our knowledge, this is the first proposal of a concept-based retrieval framework for this task.

Current concept-based video retrieval models normally operate on a fixed number of features per retrieval unit, for example the confidence scores of detectors for a number of concepts (Donald and Smeaton, 2005; Snoek and Worring, 2009). Therefore, it is difficult to extend these retrieval models to search for video segments of varying length. To solve this, the UCLM framework models a video segment as a series of shots. Furthermore, the framework uses an analogy to text retrieval and considers the frequency of a concept in a video segment for ranking, in parallel to the frequency of a term in a text document. If we know the occurrence or absence of a concept in each shot of a news item its frequency can be determined simply by counting. However, because of the varying number of shots in news items, the concept frequencies are difficult to compare. Instead, we use them indirectly by calculating the probability that a concept is produced by a news item, re-using the language modeling retrieval function from text retrieval (Hiemstra, 2001; Ponte, 1998). Due to the novelty of the task and the fact that segmentations of broadcast news videos into news items are readily available, this chapter focuses on *news items* as video segments.

In order to use the concept language modeling retrieval function, we have to cope with two problems: (1) the occurrences of the concepts in the shots of a news item are uncertain and (2) we have to select the concepts to use for retrieval because they are not necessarily named in the query text. To handle the uncertainty (1), we apply the URR framework to an uncertain document representation of concept frequencies and the language modeling retrieval function. For (2) we use the Annotation-Driven Concept Selection (ADCS), proposed in Chapter 4, which uses an annotated development collection to find useful concepts for retrieval.

The remainder of this chapter is structured as follows: in Section 6.2 the background concerning the language modeling framework in text retrieval is given. Section 6.3 describes the application of the UCLM framework to the news item retrieval problem. The experiments which show the effectiveness of our framework are described in Section 6.4. Finally, Section 6.5 ends this chapter with a summary and a discussion.

6.2 Background: Language Modeling

This section gives the background on language modeling for text retrieval and an application to the retrieval of uncertain spoken documents.

6.2.1 Language Modeling

This section describes the basic language modeling framework which is used in this chapter. The interested reader is referred to Zhai and Lafferty (2004) for a more in-depth discussion. In principle, the language modeling framework ranks documents by the probability that the query is drawn from the document:

$$P(\vec{t}|d) = \prod_{i}^{n} P(t_i|d) \tag{6.1}$$

Here, $\vec{t}$ are the query terms and the probability $P(t_i|d)$ is the *draw probability* that term t_i is drawn from the document's distribution. After the so-called smoothing, which estimates the probability $P(t_i|d)$ does the language modeling ranking function depends on term frequencies. The smoothing technique used in this chapter is the Dirichlet smoothing (Zhai and Lafferty, 2001), which estimates the term draw probability as follows:

$$P(t|d) = \frac{tf_t + \mu\ P(t|\mathcal{D})}{dl + \mu}$$

Here, tf_t is the term frequency of term t in document d, dl is the document length, $P(t|\mathcal{D})$ is the collection prior of the term and μ is the Dirichlet parameter. The retrieval function for a query is then defined as:

$$retfunc_{LM}\langle \vec{TF}\rangle(\vec{tf} : dom(\vec{TF}), dl : \mathbb{N}) = \prod_{i=1}^{n} \frac{tf_i + \mu\ P(t_i|\mathcal{D})}{dl + \mu} \tag{6.2}$$

Here, n is the number of query terms and Equation 6.2 is a common language modeling ranking function.

6.2.2 Uncertainty in Spoken Document Retrieval

Chia et al. (2008) used the language modeling framework in spoken document retrieval. It uses a document representation of expected term frequencies given the low-level features, observed by an automatic speech recognition system (Huijbregts, 2008). Note, this implicitly requires a distribution of term frequencies given this observation $P_\Omega(TF(d)|O = \vec{o})$, which is defined by Chia et al. (2008). The retrieval function is described as follows:

$$retfunc_{ETFLM}\langle \vec{O}\rangle(\vec{o} : dom(\vec{O})) = \prod_{i}^{n} \frac{E[TF_i(d)|\vec{o}] + \mu\ P(t_i|\mathcal{D})}{E[DL(d)|\vec{o}] + \mu} \simeq P(\vec{t}|d) \tag{6.3}$$

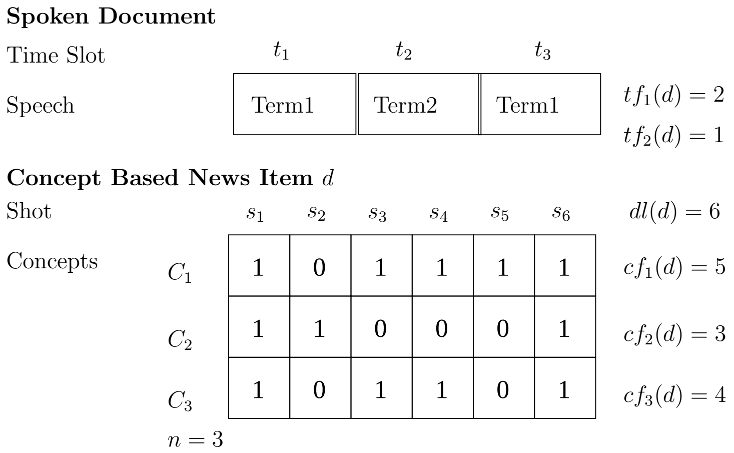
Here, $\vec{o}$ are the observations of the automatic speech recognition system and $E[TF_i(d)|\vec{o}]$ is the expected term frequency of term i given the observations $\vec{o}$. Similarly, since it is not known how many words have been said, $E[DL(d)|\vec{o}]$ is the expected document length. Because of the good performance of this approach, this chapter will use this retrieval function as a baseline in the experiments in Section 6.4 – only using expected concept frequencies instead of term frequencies.

6.3 Uncertain Concept Occurrence Language Model

6.3.1 Concept-Based News Item Representation

A broadcast news video can naturally be segmented into news items. Furthermore, these items can be subdivided into shots. Figure 6.1 shows the analogy between document representations of spoken text and concept-based video segments. The spoken document consists of three spoken words at time position $t_1 - t_3$ and the news item of six shots $s_1 - s_6$ and three concepts $\mathcal{V} = \{C_1, C_2, C_3\}$. On the right, we see the term and concept frequencies of the documents as the count of the occurrences on the left – the analogy between the proposed representation of news items and spoken documents. We denote the occurrence of concept i in shot $d.s_j$[1] as $c_i(d.s_j) \in \{0, 1\}$ where

[1] Note that, in Figure 6.1 the shorter notation s_j for the shot $d.s_j$ was used for display reasons.



Spoken Document

Time Slot	t_1	t_2	t_3	
Speech	Term1	Term2	Term1	$tf_1(d) = 2$ $tf_2(d) = 1$

Concept Based News Item d

Shot		s_1	s_2	s_3	s_4	s_5	s_6	$dl(d) = 6$
Concepts	C_1	1	0	1	1	1	1	$cf_1(d) = 5$
	C_2	1	1	0	0	0	1	$cf_2(d) = 3$
	C_3	1	0	1	1	0	1	$cf_3(d) = 4$
	$n = 3$							

Figure 6.1: *A concept-based News Item Representation and its Analogy to a Spoken Document.*

1 stands for the occurrence of the concept. Now, if we know about the occurrences of the concepts in a news item, we can express the frequency as a sum (count): $cf_i(d) = \sum_j^{dl} c_i(d.s_j)$.

6.3.2 Concept Language Models

We now describe our ranking function for concept-based news item retrieval. The basic idea behind our approach is to consider the occurrence and absence of a concept as two concept words of the language of this concept. The two conceptwords are 'occurs' or 'is absent' and instead of a single stream of terms we have multiple concept streams.

As the concept frequencies between news items are difficult to compare we consider, in parallel to language modeling (Hiemstra, 2001; Ponte, 1998), the draw probability of a concept from a concept stream of a news item. We apply Dirichlet smoothing (Zhai and Lafferty, 2004) to estimate the unknown draw probabilities for a concept C in news item d :

$$P(C|d) = \frac{cf + \mu\ P(C|\mathcal{D})}{dl + \mu} \tag{6.4}$$

Here, cf is the concept frequency of concept C in the news item d, $P(C|\mathcal{D})$ is the prior of encountering concept C in the collection $\mathcal{D}$, μ is the Dirichlet parameter, and finally dl is the document length (in numbers of shots). Note, in contrast to the approach of Chia et al. (2008), see Section 6.2.2, in our approach the document length dl always known, since we can observe how many shots a news item comprises. We can now rank news items by the probability of drawing a set of concepts independently from their concept stream:

$$retfunc_{UCLM}\langle \vec{CF} \rangle(\vec{cf} : dom(\vec{CF}), dl : \mathbb{N}) = \prod_i^n \frac{cf_i + \mu\ P(C_i)}{dl + \mu} \tag{6.5}$$

Here, $C_1, ..., C_n$ are important concepts for the information need and the equation calculates the probability of sampling these concepts from the news item d. Equation 6.5 is the concept language model, proposed by this chapter. To select these important concepts, this chapter uses the ADCL method, proposed in Chapter 4.

6.3.3 Uncertain Concept Occurrences

Until now we have considered concept-based news item search for the case of known concept occurrences. However, in reality we will only have probabilistic knowledge about the concept occurrence through the output of concept detectors. As the concept occurrence is unknown, let $CF_i(d)$ be the random variable "the concept frequency of concept i in news item d. Recall that the occurrence probability of a concept C in a shot $d.s$ is $P_\Omega(C(d.s)|o(d.s))$, which can be calculated in Section 2.3.4. With this probabilistic knowledge of a concept occurring in shots, we can determine the probability distribution over the possible concept frequencies for $CF_i(d)$ of document d. For example, the probability that concept i has a frequency of one in a news item d of document length $dl = 3$ is:

$$\begin{aligned} P_\Omega(CF_i(d) = 1|\vec{o}) &= P_\Omega(\vec{C}_i(d) = 1,0,0|\vec{o}) \\ &+ P_\Omega(\vec{C}_i(d) = 0,1,0|\vec{o}) \\ &+ P_\Omega(\vec{C}_i(d) = 0,0,1|\vec{o}) \end{aligned} \tag{6.6}$$

Here, $\vec{C}_i(d)$ is a short form for $(C_i(d.s_1), C(d.s_2), C(d.s_3))$ and the probability $P_\Omega(\vec{C}_i(d) = 1,0,0|\vec{o})$, from Equation 6.6 above, is calculated as follows:

$$\begin{aligned} P_\Omega(\vec{C}_i(d) = 1,0,0|\vec{o}) &= P_\Omega(C_i(d.s_1)|o_i(d.s_1)) \\ &\quad (1 - P_\Omega(C_i(d.s_2)|o_i(d.s_2))) \\ &\quad (1 - P_\Omega(C_i(d.s_3)|o_i(d.s_3))) \end{aligned}$$

Finally, the probability that an uncertain document representation of concept frequencies $\vec{CF}(d)$ is equal to $\vec{cf}$ can be calculated as follows:

$$P_\Omega(\vec{CF}(d) = \vec{cf}|\vec{o}(d)) = \prod_i^n P_\Omega(CF_i(d) = cf_i|\vec{o}) \tag{6.7}$$

These probabilities can be used in calculating the expected score and its variance in Section 6.3.4.

6.3.4 Retrieval under Uncertainty

This section describes how the concept language ranking function from Equation 6.5 and the uncertainty of the concept occurrences in shots are used in the UCLM framework, proposed by this chapter. As described above, the vector of the concept frequencies $\vec{CF}(d) = (CF_1(d), ..., CF_n(d))$ is the uncertain document representation of document d[2]. However, since we do not know the exact representation let $dom(\vec{CF}(d))$ be the set of all possible representations in which the news item could be. Furthermore, let $score_q$ be the concept language model score function of the current information need, see Equation 6.5. Because of the representation uncertainty, let $S(d) = score_q(\vec{CF}(d))$ be the uncertain concept language score for a document d with uncertain document representation $\vec{CF}(d)$. According to the URR framework from Chapter 3, we now define the expected score and its variance to arrive at a ranking score value for news items.

Expected Score The expected score for the concept language model score function $score_q$ is defined as:

$$E[S(d)|\vec{o}] = \sum_{\vec{cf} \in dom(\vec{CF}(d))} score_q(\vec{cf}) P_\Omega(\vec{CF}(d) = \vec{cf}|\vec{o}) \tag{6.8}$$

Here, $\vec{cf}$ is one of $|dom(\vec{CF}(d))|$ possible representations of the concept frequency representation of the document, each of which have an assigned score $score_q(\vec{cf})$. Then, the expected score is the weighted average of these scores, according to the occurrence probability of the representation $P_\Omega(\vec{CF}(d) = \vec{cf}|\vec{o})$, which can be calculated according to Equation 6.7.

Variance of the Score The second component of the URR framework is the variance of the score $S(d)$ which is, following the derivation in Equation 3.6, calculated via the expected squared score:

$$\begin{aligned} \mathrm{var}[S(d)|\vec{o}] &= E[S(d)^2|\vec{o}] - E[S(d)|\vec{o}]^2 \text{ with} && (6.9) \\ E[S(d)^2|\vec{o}] &= \sum_{\vec{cf} \in dom(\vec{CF}(d))} score_q(\vec{cf})^2 P_\Omega(\vec{CF}(d) = \vec{cf}|\vec{o}) && (6.10) \end{aligned}$$

The variance expresses how much the possible scores vary in the distribution of possible scores.

Combining the Components Finally, following the URR framework, see Section 3.3.4, news items should now be ranked by a combination of the expected score and its standard deviation:

$$RSV(d) = \underbrace{E[S(d)|\vec{o}]}_{\text{Eq. 6.8}} - b \underbrace{\sqrt{\mathrm{var}[S(d)|\vec{o}]}}_{\text{Eq. 6.9}} \tag{6.11}$$

Here, b is the risk parameter representing the risk attitude of the retrieval engine. If $b > 0$, the retrieval engine is called risk-averse. For $b = 0$ the retrieval engine is risk neutral and for $b < 0$, we call the retrieval engine risk-loving. Equation 6.11 is the direct application of the URR framework on representations of concept frequencies, the uncertain concept language model UCLM framework.

[2] For an explanation of document specific document variables see Appendix A.

Algorithm 6.1: Implementation of the sampling procedure *GenerateSamples* for video segment retrieval, derived from Algorithm 3.2.

```
Data: Collection $\mathcal{D}$,
Collection of samples $\mathcal{D}_S$,
Document Features $\mathcal{V}$,
Distribution $P_\Omega(\mathcal{V}|\vec{o})$,
Number of samples $NS$

GenerateSamples()
begin
    foreach Document d in $\mathcal{D}$ do
        for l=1 to NS do
            $d^l$ := newDoc();
            for $C \in \mathcal{V}$ do
                $cf_C(d^l) := 0$
                for i=1 to dl do
                    sm := uniform sample from [0 : 1]
                    if $sm < P_\Omega(C|o_i(d.s_j))$ then
                        $cf_C(d^l)$++;
                    end
                end
            end
            Append $d^l$ to $\mathcal{D}_S$;
        end
    end
end
```

6.3.5 Implementation

As the number of possible concept frequencies for a document representations is large and calculating the expected score, see Equation 6.8, and the expected squared score, see Equation 6.10, is computationally expensive, we apply the Monte Carlo estimation method, see (Liu, 2002), to estimate both expectations. The method is defined as follows: Let $\vec{cf}(d^1), ..., \vec{cf}(d^{NS})$ be NS random samples from the distribution of possible document representations $P_\Omega(\vec{CF}(d)|\vec{o})$. Then the expectations from Equation 6.8 and Equation 6.10 can be approximated by:

$$E[S(d)|\vec{o}] \simeq \frac{1}{NS} \sum_{l=1}^{NS} score_q(\vec{cf}(d^l)) \tag{6.12}$$

$$E[S(d)^2|\vec{o}] \simeq \frac{1}{NS} \sum_{l=1}^{NS} score_q(\vec{cf}(d^l))^2 \tag{6.13}$$

Because the standard error of the Monte Carlo estimate is in the order of $1/\sqrt{NS}$ we can achieve a relatively good estimate already with few samples. Algorithm 6.1 shows how such random samples for news items are generated. For each shot j and each concept i a sample of a concept occurrence $C_i(d.s_j)$ is obtained. First, a uniformly distributed random number sm from the interval $[0:1]$ is generated. Second, if the sm was smaller than $P_\Omega(C_i(d.s_j)|o(d.s_j))$ we assume concept occurrence and add one to the concept frequency $cf_i(d^l)$ of the concept i in the sample l of news item d. After processing all concepts for all shots of a news item, we store this sample document representation in a separate collection $\mathcal{D}_S$.

Algorithm 6.2 shows how a query is processed. First, the ADCS method is used to determine important concepts for this query. Then a new score function is derived from the concept language retrieval function, see Equation 6.5. Afterwards, for all generated samples of a news item d a score is calculated using the score function and the known document representation of the sample. The expected score ES' and expected square score $ES2'$ are calculated according to Equation 6.12 and Equation 6.13. Finally, the score's standard deviation $\sqrt{ES2' - ES'^2}$ is calculated and the document is ranked according to Equation 6.11 using the risk parameter b, which is a system parameter of the retrieval engine.

6.4 Experiments

This section describes experiments that were performed to evaluate the benefits of the UCLM framework.

6.4.1 Experiment Setup

The experiments are based on the TRECVid 2005 collection which comprises 180 hours of Chinese, Arabic and English broadcast news (Smeaton et al., 2006). NIST announced the automatic shot segmentation from Petersohn (2004) as the official shot boundary reference, defining a total of $45,765$ shots. For the segmentation of the videos into news items, we used the results from Hsu et al. (2006), which looked for the anchorperson in the video, to determine a news item change. This segmentation resulted in $2,451$ news items of an average length of 118 seconds. We associate a shot with a news item, if it starts within the time interval of the news item. This results in an average of 17.7 shots per news item.

Because of the novelty of our approach there is no standard set of queries for this search task. Therefore, we decided on using the 24 official queries from TRECVid 2005, replacing the "Find shot of ..." with "Find news items about ...". Furthermore, we assume that a news item is relevant to a given information need, if it contains at least one relevant shot (which can be determined from the relevance judgments from NIST for the TRECVid 2005

Algorithm 6.2: Implementation of *retrievalrun* procedure for video segment retrieval, derived from Algorithm 3.2.

```
Data: Collection $\mathcal{D}$,
Collection of samples $\mathcal{D}_S$,
Query Representation $\vec{QF}$,
Risk Parameter $b$,
Retrieval model UCLM=$(selectNweight_{ADCS}(), retfunc_{UCLM}\langle\rangle)$,
Number of samples $NS$
$retfunc_{UCLM}\langle \vec{CF}, DL\rangle(\vec{cf} : dom(\vec{CF}), dl : \mathbb{N}) = \prod_i^n \frac{cf_i + \mu\, P(C_i)}{dl + \mu}$
$retrievalrun(\vec{qf} : dom(\vec{QF}))$
begin
    // Score Function Definition
    $(\vec{C}, w) := selectNweight_{ADCS}(\vec{qf})$
    // concept frequency features corresponding to selected concept $\vec{C}$
    $\vec{CF} := (CF_{c_1}, \ldots, CF_{|c_n|})$
    $score_q := new\ retfunc_{UCLM}\langle \vec{CF}, dl\rangle$
    // Matching and Combine
    foreach $Document\ d\ in\ \mathcal{D}$ do
        $ES' := 0$ // $\simeq E[S(d)|\vec{o}(d)]$ ;
        $ES2' := 0$ // $\simeq E[S(d)^2|\vec{o}(d)]$;
        foreach $Sample\ Document\ d^* \in \mathcal{D}_S\ of\ d$ do
            $s = score_q(\vec{cf}(d^*), dl(d^*))$;
            $ES' += s$;
            $ES2' += s^2$;
        end
        $ES' = ES'/NS$;
        $ES2' = ES2'/NS$;
        Append $(d, ES' - b\sqrt{ES2' - ES^2})$ to $ranking$;
    end
    return $sort(ranking, ranking.score\ DESC)$
end
```

collection). We argue that for most queries this is realistic since the user is probably searching for the news item as a whole, rather than for shots within the news item. A similar assumption is made during the creation of relevance judgments for the text retrieval workshop TREC: here, a document is relevant is relevant if a part of it is relevant.

We used the vocabulary of 101 concepts and the corresponding detector set from the MediaMill challenge experiment for our experiments (Snoek et al., 2006). The reason for this is that it is a frequently referenced stable detector set with good performance on the mentioned collection. We use the Annotation-Driven Concept Selection method, proposed in Chapter 4, to select important concepts for a query. Here, the textual query was first executed by the general purpose text retrieval engine PF/Tijah (Hiemstra et al., 2006) on a textual representation of the development collection and the first documents in the ranking, before a cut-off value of m, are assumed to be relevant. We then use the first n concepts with the highest estimated Mutual Information as the concepts for this query. We set the parameter to a cut-off value of m=150, since this resulted in good performance for video shot retrieval.

The UCLM framework was compared to four other approaches from the uncertainty classes UC1-UC4 discussed in Section 2.4. As the score-based approaches for (UC1+UC2) are defined on fixed numbers of features we use the average probability of each considered concept as the score for this concept:

$$o_i(d) = \frac{\sum_j P_\Omega(C_i(d.s_j)|o(d.s_j)))}{dl}$$

Here, $o_i(d)$ is the normalized average score of concept C_i. The considered approaches were:

(1) CombMNZ which multiplies the scores if they are not zero (Aslam and Montague, 2001)

(2) Borda-Count which considers the rank of the average score (Donald and Smeaton, 2005).

(3) Best-1, which ranks the news items by the concept language model score of the most probable representation. To be more concrete, a concept in a shot was counted if the probability of the concept was above 0.5. The resulting concept frequencies were then used to calculate the concept language model score described in Equation 6.5

(4) We used an approach similar to the one from Chia et al. (2008), termed the expected concept frequency language model ECFLM. The expected

Retrieval Model	Number of Concept n	MAP	P10
CombMNZ	10	0.105	0.045
Borda-Count	1	0.090	0.000
Best-1	5	0.094	0.245
ECFLM	10	0.192	0.287
UCLM	10	0.214*	0.291

Table 6.1: *Results of comparing the proposed UCLM framework against four other methods described in related work. *: The improvement of the UCLM framework has been tested for significance using a two-sided, paired Wilcoxon signed rank test with a significance level of* 0.05 *against all other methods.*

concept frequency is calculated by the following:

$$\begin{aligned} E[CF_i(d)|\vec{o}] &= \sum_{j=1}^{dl} \sum_{c \in dom(C)} c \; P_\Omega(C_i(d.s_j) = c | o_i(d.s_j)) \\ &= \sum_{j=1}^{dl} P_\Omega(C_i(d.s_j) | o_i(d.s_j)) \end{aligned}$$

Here, $E[CF_i(d)|\vec{o}]$ is the expected concept frequency and $P_\Omega(C_i(d.s_j)|o_i(d.s_j))$ is the occurrence probability of concept C_i in shot $d.s_j$. The documents were ranked by the Equation 6.5 using the expected concept frequency as the actual concept frequency:

$$retfunc_{ECFLM}\langle\vec{O}\rangle(\vec{o} : dom(\vec{O}), dl : \mathbb{N}) = \prod_{i}^{n} \frac{E[CF_i(d)|\vec{o}] + \mu \; P(C_i|\mathcal{D})}{dl + \mu}$$

6.4.2 Comparison to other Methods

Table 6.1 shows the result of the comparison of the described methods with the proposed UCLM framework. The first column after the method names indicates the number of concepts under which each method performed the best. We see that the method Borda-Count, CombMNZ and Best-1 from the uncertainty classes UC1 till UC3 perform much worse than the two methods which include multiple possible concept frequencies. For our method we used NS=200 samples, a Dirichlet prior of μ=60, and a risk factor b=-2. To rule out random effects, we repeated the run ten times and report the average. The improvement of the UCLM method against all other methods was tested for significance using a two-sided, paired Wilcoxon signed rank test with a significance level of 0.05. The search performance of the UCLM method is 0.214 MAP.

6.4.3 Study of Parameter Values

Figure 6.2 shows the result of a study over the two most important parameters in the UCLM framework. For both studies we repeat each run ten times, to rule out random effects.

Figure 6.2 (a) shows the sensitivity of the UCLM framework over the number of samples. We see that even with few samples (NS=50) the performance is better than the ECFLM method. As usual for a Monte Carlo estimator, the precision increases in line with the square root of the number of samples. After NS=250 samples we barely see any further improvement.

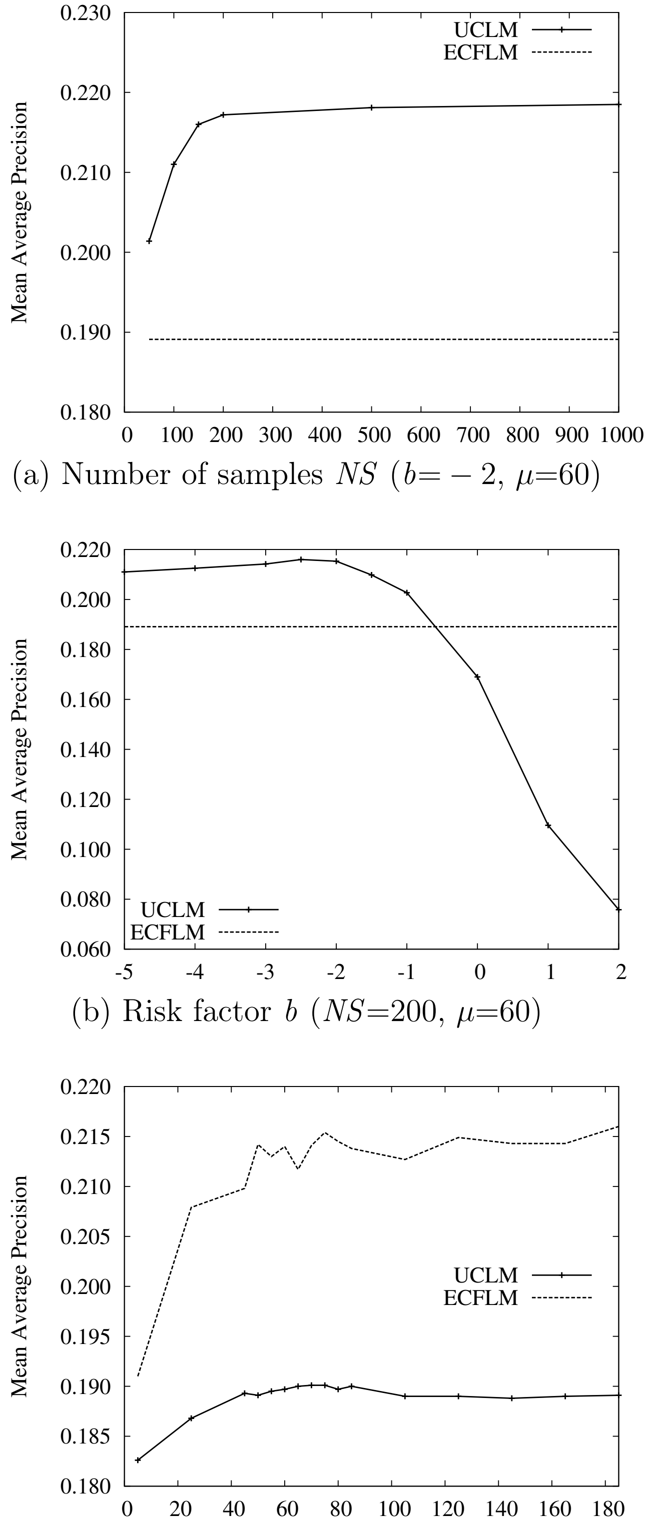
Figure 6.2 (b) shows the behavior of our model for changes of the risk parameter b. We see, with values of $b > -1$ the UCLM method performs worse than the ECFLM method. The reason for this potentially is that the concept detectors show a low performance and therefore, the variance of the concept frequencies can be high. A risk-averse engine will always rank documents with low expected scores above documents with slightly lower expected scores although the latter might have a higher chance of having a higher actual score, because of a higher score variance. Since the performance of concept detectors is still low, this suggests that a risk-loving attitude increases the search performance in this experiment.

Figure 6.2 (c) shows the search performance influence of changes of the Dirichlet parameter μ. For all values of μ the proposed UCLM method shows a better search performance than the ECFLM method.

6.5 Summary and Discussion

This chapter proposed a ranking framework for longer video segments than the commonly assumed retrieval unit of a video shot, termed the Uncertain Concept (Occurrence) Language Model (UCLM) framework. Because of the novelty of the task we focused on the search for news items, a particular segment type. We found that current shot-based retrieval models are hard to adapt to longer video segments. Therefore, we proposed a new retrieval function, a concept-based language model, which ranks a news item with known concept occurrences by the draw probability of important concepts. However, since we only have probabilistic knowledge about the concept occurrences we applied the URR framework from Chapter 3 to model the uncertainty of concept frequencies to rank by the expected concept language model score plus the associated risk a retrieval engine takes to rank a document by this score, represented by the standard deviation of the score.

We have shown that the UCLM framework performs better than retrieval models that take only the confidence scores, their ranks, or the most probable concept frequency document representations into account. We have also shown that the UCLM framework, which considers the expected score of a concept-based language model, performs significantly better than an adapted method from spoken document retrieval, which takes the expected concept



(a) Number of samples NS ($b = -2$, μ=60)

(b) Risk factor b (NS=200, μ=60)

(c) Dirichlet parameter μ (NS=200, $b = -2$)

Figure 6.2: *Robustness study of parameter settings.*

frequency and only then applies the concept-based language model.

Finally, we have shown that the performance behavior of the UCLM framework is stable over all its parameters.

Chapter 7

Detector Simulation

This chapter is based on Aly and Hiemstra (2009a).

7.1 Introduction

Content-based video retrieval currently focuses mainly on the improvement of concept detectors (Snoek and Worring, 2009). On the other hand, there is research on developing retrieval models to combine the output of the concept detectors to answer the information needs of users. However, currently the performance of such retrieval engines still often prohibits their application in real life. Clearly, the performance of the overall retrieval engine heavily depends on the detector performance. Therefore, it is desirable to answer the research question $Q5$, see Section 1.5: *How can we predict whether improved concept detection will make a current concept-based retrieval engine applicable to real-life applications in the future?* This chapter investigates the application of a Monte Carlo Simulation approach to answer this question.

Hauptmann et al. (2007) were the first to use a simulation-based approach to predict the achievable performance of concept-based video retrieval engines. In this work, noise is introduced into the known occurrences and absences of concepts by randomly flipping their states. Therefore, detectors are assumed to be binary classifiers which only differentiate between concept occurrence and absence. While assuming binary classifiers is useful to study the general applicability of concept-based retrieval, most retrieval engines today employ confidence scores or a probability measure based on this score as document representations, see Chapter 2. The reason is that errors in binary classifications are frequent and the information of "shot x contains concept y with a confidence of z" needs to be exploited. For example, the concept *US-Flag* is probably useful for answering the query "President Obama". However, the corresponding detector might never classify a shot as containing the concept *US-Flag* but may find few shots more likely to contain *US-Flags* than others, which could be exploited. Therefore, the simulation approach in this chapter generates confidence scores for each shot and a concept vocabulary which can then be transformed into probability measures as well as classifications.

The proposed approach in this chapter follows the Monte Carlo Simulation approach (Metropolis and Ulam, 1949) to predict the search performance of retrieval engines when the detector performance increases. The simulation approach requires a function which calculates a quantity we are interested in on a given set of inputs. In our case this function will be the mean average precision (MAP) of a retrieval engine and the inputs are the confidence scores of the collections. The application of the Monte Carlo Simulation approach allows us to split the broad research question Q5 into two sub-questions:

Q5.1 *How can we simulate the improvement of concept detectors?* In order to answer this question, we assume that confidence scores of detectors are independent from each other. Furthermore, we make the assumption that these confidence scores are normally distributed in the set of shots where the concept occurs and likewise where they are absent (the positive and the negative class). This assumption is supported by studies of actual detector outputs in this chapter and by Hastie and Tibshirani (1996). Therefore, our probabilistic model consists of the parameters for two Gaussian distributions for each detector for a concept vocabulary.

Q5.2 *What search performance can we expect from a retrieval engine for a given detector model?* In order to answer this question, we use the probabilistic model and a collection with known concept occurrences to generate a set of randomized confidence scores. On this output, we then execute a retrieval run using a given retrieval engine and subsequently calculate the search performance in terms of MAP. This process is repeated several times to calculate the expected mean average precision of the retrieval engine given the probabilistic model.

Having the answer to these two questions, we then gradually change the parameters of the model to improve the detector performance and investigate the effect on the expected search MAP. From the development of the expected search performance compared to the detector performance we can predict the answer to research question Q5.

The remainder of this chapter is structured as follows: in Section 7.2 we give an overview of the Monte Carlo Simulation method and an overview of related work which evaluates multimedia retrieval systems. Section 7.3 describes the probabilistic model which is used to simulate the detectors. In Section 7.4, we investigate the results of the simulation on a collection with concept annotations and relevance judgments. Section 7.5 ends this chapter with a summary and a discussion.

7.2 Background: Simulation and Performance Prediction

7.2.1 Monte Carlo Simulation

This chapter proposes a simulation approach based on Monte Carlo Simulations (Metropolis and Ulam, 1949). In the literature, the term Monte Carlo Simulation is used for a variety of different methods. Here, we use it for a general procedure to calculate the expected value of a function (here, the performance in terms of MAP) given the probabilistic model of the inputs. A Monte Carlo Simulation can be described in the following steps:

(1) The definition of a probabilistic model of the inputs to the simulation, our case the confidence scores and their distribution based on their class.

(2) Random generation of a concrete set of inputs using the model, a set of concrete confidence scores in our case.

(3) Execution of the function using the generated inputs, in our case the calculation of the search performance of a retrieval model in terms of MAP.

(4) Repetition of (2) and (3) to produce multiple results.

(5) Average the results of the individual computations into the final result.

The results of this simulation is guaranteed to converge with an increasing number of repetitions to the expected function value (performance measure), based on the probabilistic model.

7.2.2 Search Performance Prediction

Simulations which analyze the effects of recognition performance on search performance have been used in other sub-fields of content-based multimedia retrieval. Croft et al. (1992) use simulations to determine the effects of word-error-rates in optical character recognition systems on the search performance. Witbrock and Hauptmann (1997) simulate a varying word-error-rate of an automatic speech recognition system, to investigate its influence on the search performance of a spoken document retrieval engine.

Hauptmann et al. (2007) were the first to use a simulation-based approach to investigate achievable concept-based search performance. In their work, a detector is assumed to be a binary classifier. As a retrieval function they use a linear combination of concept occurrences: $score_q(d) = \sum_i w_i f_i(d)$. Here, $score_q(d)$ is the retrieval score of shot d, w_i is a concept specific weight and $f_i(d) \in \{-1, 1\}$ is the label of concept i in shot d. The weights w_i are independently set for each query. The weight setting which optimizes the average precision is found by solving a bounded constrained global optimization problem (Yan and Hauptmann, 2003). The retrieval performance with realistically set weights is assumed to achieve 50% of the performance with optimal settings. Concept labels of shots are randomly flipped until the precision-recall break-even point is reached. We argue that this approach can be improved because current retrieval engines use confidence scores and a uniform break-even precision-recall point assumes the same performance from all detectors which is unrealistic.

Similar to the approach in this chapter, Toharia et al. (2009) simulate confidence scores to study the usefulness of concept-based retrieval. A concept from an annotated collection is assumed to have a score of -1 if it is absent and 1 if it occurs. For the simulation, noise is introduced by adding or subtracting to a certain percentage of P shots a value A, which improves or decreases the performance of the detector set. As a retrieval function a weighted sum of the confidence scores is assumed where the weights are determined by users. The simulation is carried out by varying the percentage P from 0 to 0.5 and A from -0.5 to 0.5. While this approach also simulates the influence of confidence scores on the search performance, it does not consider that the confidence scores for concepts, which are absent, could be higher for some shots where the concept occurs. Therefore, the detector MAP is always 1.00. Our simulation improves upon this.

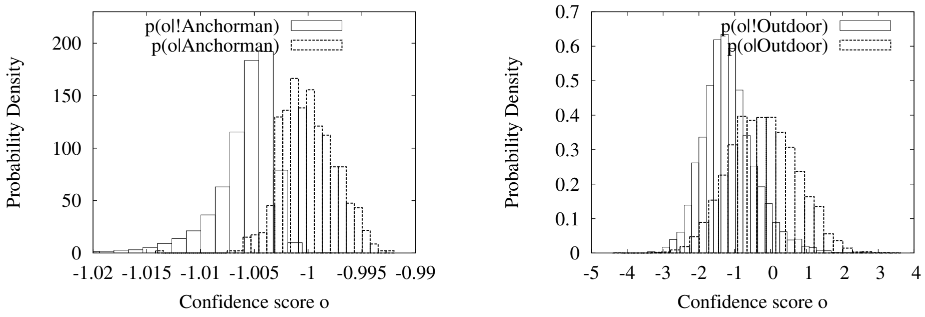
There are also other aspects than the detector performance which influence the search performance of a concept-based retrieval system which can also be simulated but are not covered in this chapter: Christel and Hauptmann (2005) investigate the general helpfulness of single concepts to retrieval. Furthermore, Snoek and Worring (2007); Hauptmann et al. (2007) study the effects of concept vocabulary size on the search-performance by randomly including or excluding a growing number of concepts.

7.3 Detector Model and Simulation

In this section we describe the probabilistic model proposed in this chapter, and the simulation process.

7.3.1 Detector Model

In this section we describe the probabilistic model of confidence scores, which will be later used for the randomization of confidence scores. Figure 7.1 shows the confidence score histograms of the two concepts *Anchorman* and *Outdoor* for the positive and the negative class from a baseline detector set, described by Snoek et al. (2006). The different score ranges and the resulting probability density magnitudes are caused by the detector's ability to discriminate between positive and negative examples. We propose that both the densities for the positive and negative class of both concepts have roughly a Gaussian



Anchorman Outdoor

Figure 7.1: *Confidence score distributions of two concepts of the MediaMill detector set (Snoek et al., 2006).*

shape. This shape was also proposed by Hastie and Tibshirani (1996) for the distribution of decision scores for general classifiers. We also conducted a χ^2 goodness-of-fit test, see Taylor (1996) for a definition, to assess the fit of these distributions. The test revealed that 31 of the 101 detectors in the vocabulary can be accepted as Gaussian at a significance level of 0.05. Out of the 31 concepts which were accepted, 22 had more than 800 training examples, which suggests that the Gaussian shape would also become evident for other concepts if we had more training examples. Furthermore, Sangswang and Nwankpa (2003) argue that a non-perfect fitting shape of a model only increases the variance of the Monte Carlo Simulation, but still allows a trustworthy estimation of the expected search performance.

Given these observations, we define a probabilistic model of a detector set: we assume that the confidence scores of different detectors for a single shot are independent from each other and that they are normally distributed in the positive and the negative class. Each concept C has a different prior probability $P(C)$. To keep the probabilistic model simple, we assume that all concepts share the same mean μ_1 and standard deviation σ_1 for the positive class plus the mean μ_0 and the standard deviation σ_0 for the negative class. Note that this assumption is strong and certainly does not hold in reality, see Figure 7.1. However, as here we focus on the principle influence of the detectors on the search performance we leave the exploration of a more realistic model that investigates different parameter settings for each detector, to future work. Also, while the investigation of different means and deviations is important, we argue that the intersection of the areas under the probability density curves has a much higher influence on the performance than the absolute ranges of the confidence scores. The smaller the area of the intersection the better the detector is. Our model can adequately simulate this effect by either moving the means apart or by varying the standard deviation of the positive and negative class.

Figure 7.2 shows the model of a single detector. We also plot the posterior probability of observing the concept given the confidence score using two

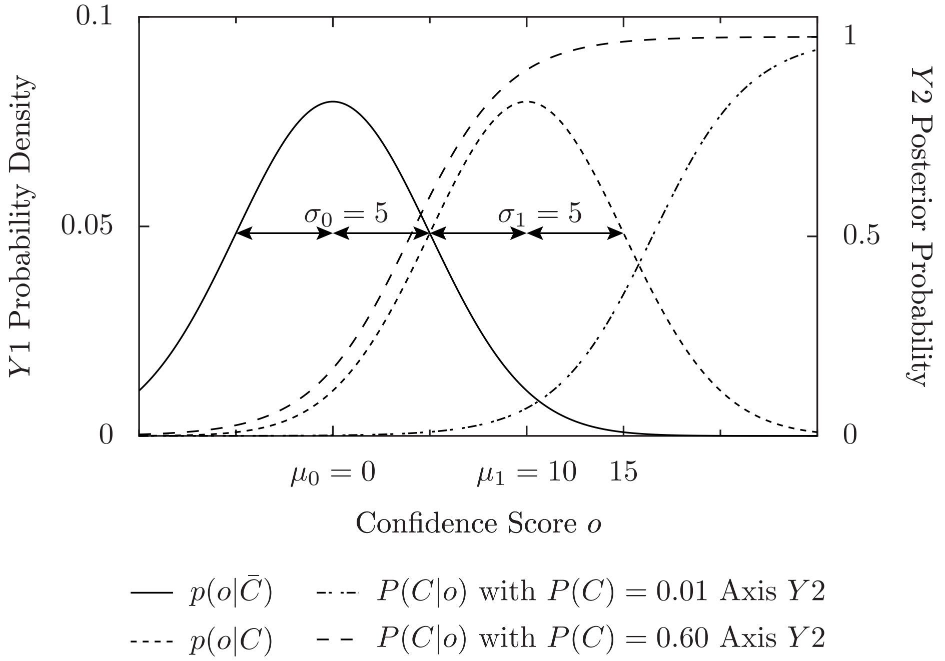


Figure 7.2: *Probabilistic detector model consisting of two Gaussians for the positive and negative class together with two possible posterior probability functions for different priors.*

different priors, one of $P(C)$=0.01 and one for $P(C)$=0.50. Considering a confidence score of o=15 the posterior probability for a concept with the prior of 0.50 is close to certainty ($P_\Omega(C|o) \simeq 1$) while for a concept with a prior of 0.01 it is approximately undecided (50%) - with all other parameters equal. Therefore, our model does not have the limitation that all detectors have the same performance as assumed by Hauptmann et al. (2007).

7.3.2 Posterior Probability

As noted by Platt (2000), the assumption of two Gaussians for the negative and positive class can lead to unwanted effects for the posterior probability function, namely that the function can be non-monotonic. Figure 7.3 shows the posterior probability functions of two hypothetical concept detectors defined by the standard formula for posterior probabilities:

$$P_\Omega(C|o) = \frac{p(o|C)P_\Omega(C)}{p(o|C)P_\Omega(C) + p(o|\bar{C})P_\Omega(\bar{C})}$$

We see that with a standard deviation of σ_1=15, the posterior probability increases for confidence scores smaller than o=-3. Furthermore, the posterior probability function with $\sigma_1 = 2$ assigns a posterior probability of practically 0 to shots with confidence scores higher than $o = 20$. This contradicts our intuition and the definition of SVM based detector (where the positive and the negative class should be linearly separable, see Section 2.2.2). To prevent

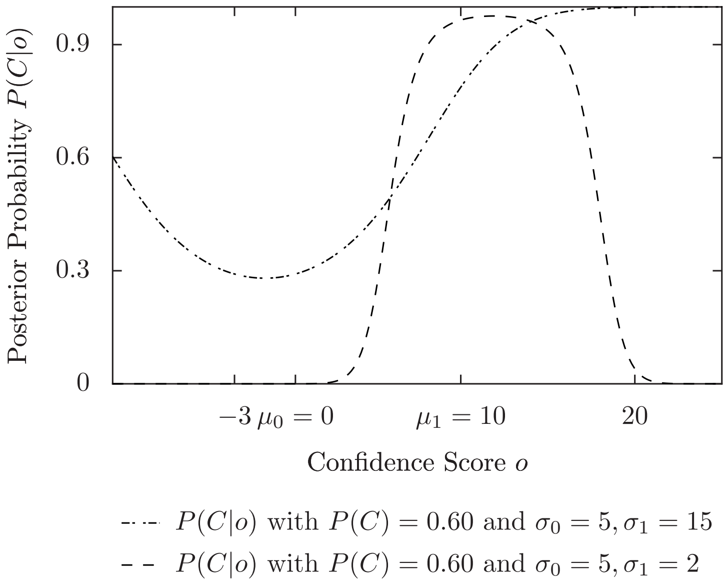


Figure 7.3: *Non-monotonic posterior probability functions, resulting from using two Gaussians.*

this effect we use an improved version of the algorithm from Platt (2000), suggested by Lin et al. (2007), to fit the parameters of a sigmoid function to the confidence scores of a set of training examples. The sigmoid function is defined as follows, see also Section 2.3.4:

$$P(C|o) = \frac{1}{1 + \exp\left(A\, o + B\right)} \tag{7.1}$$

Here, A and B are the two parameters of the sigmoid function. Note that the algorithm from Lin et al. (2007) depends on the number of training examples, retrieval models which depend on the probabilistic output of Equation 7.1 could suffer from a poorly fitted posterior function. To investigate the influence of the quality of the fit on the search performance, we use S hypothetic training examples for the fitting process and randomly generate $\lceil S\ P(C) \rceil$ confidence scores from the positive class and $S - \lceil S\ P(C) \rceil$ from the negative class of concept C. The results of this investigation can be found in the Experiment in Section 7.4.6.

7.3.3 Simulation Process

In this section, we describe the actual simulation process which is described in pseudo-code in Algorithm 7.1. The algorithm uses an annotated collection (which carries 0/1 labels for each concept in each shot). The input parameters of the algorithm are the means μ_0, μ_1 and standard deviations σ_0, σ_1 of the positive and the negative class and the number of training examples S to fit the posterior function. A Gaussian distribution with mean μ and standard deviation σ is denoted as $N(\mu, \sigma)$.

Algorithm 7.1: Algorithm for a simulation run. N: Number of Repetitions, S: Sample size for sigmoid fitting. $\mu_0, \sigma_0, \mu_1, \sigma_1$: Model parameters.

```
Data: Annotated Collection $\mathcal{D}$, Vocabulary $\mathcal{V}_C$
Input: $NR, S, \mu_0, \sigma_0, \mu_1, \sigma_1$
Result: Randomized collection

// Randomize Prior Estimate
foreach Concept $C$ in Vocabulary $\mathcal{V}_C$ do
    Calculate $P(C)$ from annotations in $\mathcal{D}$
    generate $\lceil S\,P(C) \rceil$ positive training examples from $N(\mu_1, \sigma_1)$
    generate $S - \lceil S\,P(C) \rceil$ negative training examples from $N(\mu_0, \sigma_0)$
    determine $A_C$ and $B_C$ according to Lin et al. (2007), given the
    training examples
end
// Randomize Detection Output
for Repetition $i \in [1..NR]$ do
    foreach Shot $s$ in Collection $\mathcal{D}$ do
        foreach Concept $C$ in Vocabulary $\mathcal{V}_C$ do
            if $c(s) = 1$ // Concept $C$ occurs in $s$ then
                draw $o$ from $N(\mu_1, \sigma_1)$
            else
                draw $o$ from $N(\mu_0, \sigma_0)$
            end
            // Calculate Posterior according to Platt (2000)
            $P(C|o) = \frac{1}{1+exp(A_C o + B_C)}$
            // Transform to Binary Value
            if $P(C|o) > 0.5$ then
                $C = 1$
            else
                $C = 0$
            end
        end
    end
    Calculate Detector Performance $DMAP_i$
    Search Run with Retrieval Model
    Calculate Search Performance $SMAP_i$
end
Report Detector and Search Map
$DMAP = \frac{\sum_i DMAP_i}{NR}$    $SMAP = \frac{\sum_i SMAP_i}{NR}$
```

Name	Videos	Shots
tv05d	$141 - 277$	$43,907$
mm.dev	$141 - 238$	$30,630$
mm.test	$239 - 277$	$13,277$

Table 7.1: *Collection statistics for TRECVid 2005 collection used in the simulations.*

From the annotated collection we calculate the prior probability $P(C)$ of the collection. We then generate confidence scores for the positive and the negative class using the prior probability and a total of S training examples. Now, we use the algorithm described by Lin et al. (2007) to fit the sigmoid posterior probability function to the generated training examples. After the determination of the sigmoid parameters we iterate over all shots in the annotated collection. For each shot we determine for each concept in the vocabulary whether it occurs and draw a random confidence score o from the corresponding normal distribution. Afterwards, we calculate the posterior probability of this concept in the shot using the sigmoid function with the previously determined parameters A_C and B_C. For retrieval models which use binary classifications we assume a positive occurrence if the posterior probability is above 0.5. This is justified by decision theory, see for example Bather (2000).

After the randomization, we determine the detector MAP of the detector output ($DMAP_i$). We then execute a search run for each retrieval model using the randomized collection. We then evaluate the resulting ranking using relevance judgments to obtain the search MAP ($SMAP_i$) for this run. This process is repeated NR times to rule out random effects and the expected detector performance (DMAP) and search performance (SMAP) are calculated.

7.4 Simulation Results

In the following we describe the results of the simulation runs.

7.4.1 Simulation Setup

We perform all simulations on the TRECVid 2005 development collection (tv05d) because, to the authors' knowledge, this is the only suitable collection where both concept annotations and relevance judgments are available[1]. We use the 24 original queries from TRECVid 2005 (Smeaton et al., 2006). To prevent over-fitting when performing realistic concept selections we divide

[1] The relevance judgments on the development collection were kindly provided by Rong Yan formerly at Carnegie Mellon University (Yan and Hauptmann, 2007)

the collection according to the MediaMill Challenge setting (Snoek et al., 2006) into the sub-collections mm.dev and mm.test. The statistics for the collections are summarized in Table 7.1. We use two concept vocabularies in our simulation to ensure that our results are not vocabulary specific. First, the MediaMill vocabulary (Snoek et al., 2006) which comprises 101 concepts. Second, the Vireo vocabulary (Jiang et al., 2010) is used which is a subset of the LSCOM vocabulary (Naphade et al., 2006) and comprises 374 concepts.

We use a Java-based (pseudo) random number generator[2] which implements a standard algorithm described by Press et al. (1992). For every simulation run we use a new seed for the generator to ensure a high quality of randomness, which is beneficial for Monte Carlo Simulations. To reduce random effects in the results we repeat every simulation run NR=25 times. The simulation results did not change anymore after this number of repetitions. In the following we use the common term MAP, instead of emphasizing every time that the number is actually obtained as an average over 25 runs. Note that this chapter is based on the generated confidence scores from our previous work (Aly and Hiemstra, 2009a) to produce the same results and ensure a high degree of randomness by using changing seeds. However, in follow-up work (Aly and Hiemstra, 2009b) we present software which generates confidence scores for an arbitrary collection using a fixed seed, which has the advantage that the generated confidence scores can be reproduced and the simulation experiments can be repeated.

To give an indication of the quality of the detectors we report the achieved detector MAP on the provided annotations. We used the same standard cut-off level of $2,000$ as done for the High Level Feature task in TRECVid (Smeaton et al., 2006) to maintain comparability to other results. However, this cut-off level sometimes leads to counterintuitive results because some frequent concepts occur more than $2,000$ times and consequently even a perfect detector would have an average precision of less than 1.0. Therefore, in such cases we assumed a maximum of $2,000$ shots in which the concept occurred.

Table 7.2 gives an overview of the retrieval models that were simulated in this chapter. The PMIWS, BIM and PRFUBE model require the occurrence probability of a concept given relevance $P(C|R)$ as a weight. For Borda-Count we assume that the Mutual Information of a concept is the ideal weight for this concept. The mutual information can be calculated using the three parameters $P(C)$, $P(R)$ and $P(C|R)$, see Section 4.3.3. To supply these weights and select concepts we use two alternatives. First, we perform one experiment using oracle weight settings, where we use the concept annotations and relevance judgments and determine the optimal weights by counting. Second, we perform another experiment of a realistic scenario where we use the Annotation-Driven Concept Selection method, proposed in Chapter 4, which is based on an annotated development collection. To use

[2] `http://www.ee.ucl.ac.uk/~mflanaga/java/PsRandom.html`

Video Shot Retrieval		
Ret. Func.	Description	Definition
PMIWS	Pointwise Mutual Information Weighting Scheme (see Sec. 2.4.2)	$\sum_i \log(\frac{P(C_i\|R)}{P(C_i)})P(C_i\|o_i$
Borda-Count	Rank Based (see Sec. 2.4.3)	$\sum_i w_i \; rank(P(C_i\|o_i))$
BIM	Binary Independence Model (see Sec. 2.4.4)	$\sum_i c_i' \log\left(\frac{p(1-q)}{q(1-p)}\right)$
PRFUBE	Probabilistic Ranking Framework for Uncertain Binary Events	see Chapter 5.

Video Segment Retrieval		
Ret. Func.	Description	Definition
CombMNZ	Multiply non-zero (see Sec. 2.4.2)	$\prod_i P(C_i\|o_i)$
Best-1	Concept Language Model using classifications (see Sec. 6.4)	$\prod_i \frac{cf_i + \mu \; P(c_i)}{dl + \mu}$
ECFLM	Concept Language Model using Expect Concept Frequencies (see Sec. 6.4)	$\prod_i \frac{E[CF(d)\|\vec{o}] + \mu \; P(C_i\|\mathcal{D})}{dl + \mu}$
UCLM	Uncertain Concept Occurrence Language Model	see Chapter 6.

Table 7.2: *Overview of retrieval functions (Ret. Func.) used in the simulations ($\mu = 60$,$p = P(C|R)$,$q = P(C|\bar{R})$).*

this estimation method without introducing over fitting effects we use the collection mm.test for weight estimation and later execute the search only on mm.dev. We also have to set the number of concepts which should be used for the search. As this is not the focus of this chapter we try multiple numbers of concepts with a maximum of 20 together with the results of using all concepts in the vocabulary.

7.4.2 Simulation Parameter Variation

As our goal is to study the influence of the detector performance over the different model parameters we vary them piecewise to see the effect of each parameter on the overall search performance. The methods for video segment retrieval are comparatively new and therefore we mainly focus on simulating video shot retrieval models. In the following we describe each kind of variation and the characteristics of the set of detectors resulting from it:

- We increase the mean of the positive class. In reality, this is the case if the low-level features become increasingly discriminative and the detector performance increases.

- We increase the standard deviations of the positive class. Detectors with a higher standard deviation have more extreme results. For many shots where a concept actually occurs the detector is nearly certain of the occurrence (has a high confidence score) while for many other shots the detector has a low confidence score.

- We increase the number training examples for fitting the sigmoid posterior probability function. This investigates the influence of the fit quality, caused by a small number of training examples, on the search performance.

- For video segment retrieval, we increase the mean of the positive class to investigate how the retrieval models behave under improved detector performance.

7.4.3 Model Coherence

In this section, we exemplarily investigate the coherence of the proposed probabilistic model with the MediaMill Challenge detector set by Snoek et al. (2006). We first fit the model parameters to the confidence scores of the detector set. We expect that the average detector performance is close to the detector performance of the real detectors. However, the search performance of the simulation is not necessarily equal to the real search performance, because of the random distribution of confidence scores in relevant shots. On the other hand, the real search performance should also not be too far from the search performance produced by the model.

Measure	Expected Result	Simulation Max	Real Detectors
Detector MAP	0.13	0.16	0.15
Search MAP	0.06	0.11	0.10

Table 7.3: *Simulation results for investigating the model coherence.*

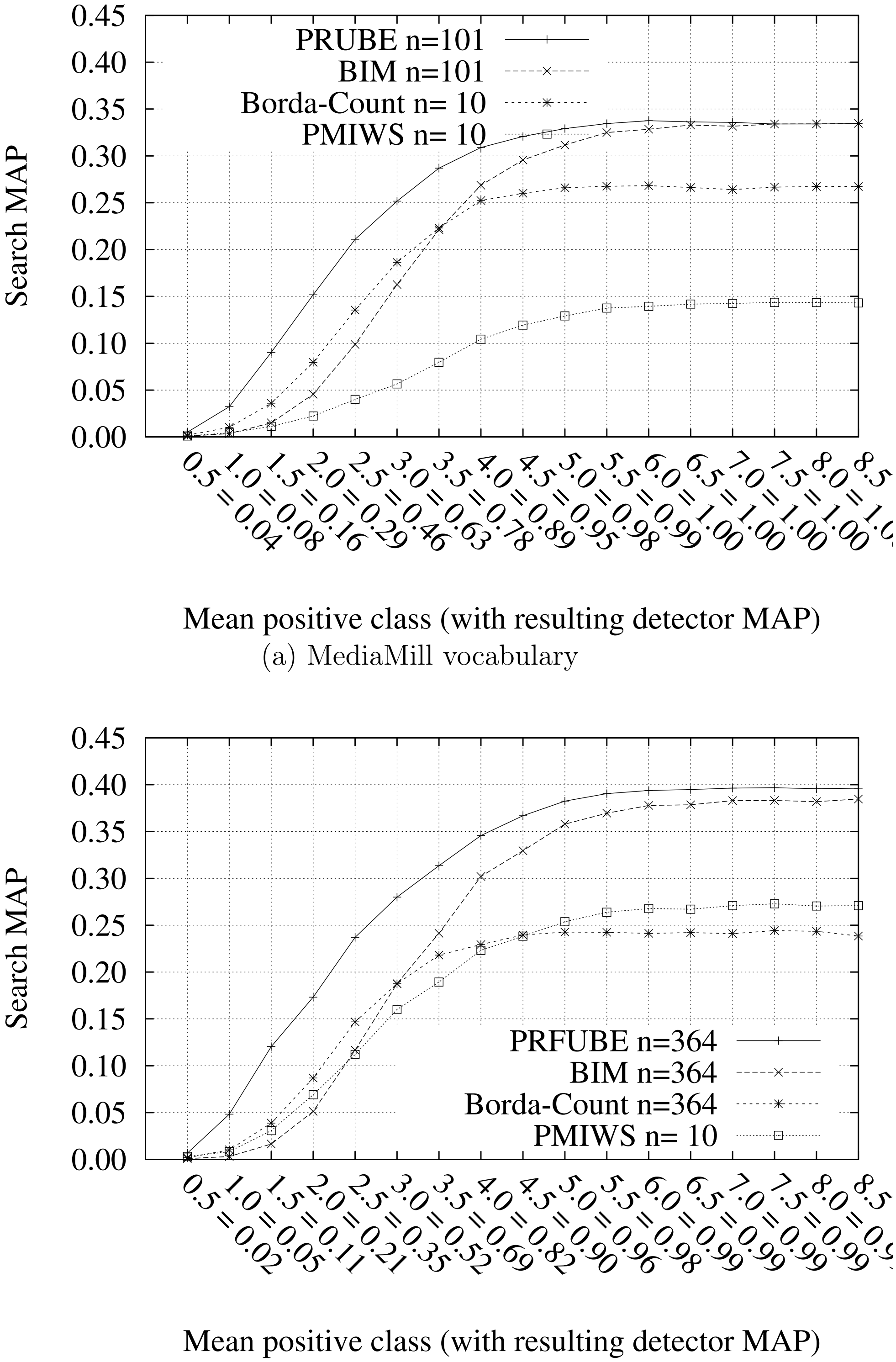
First, we train detectors for the MediaMill vocabulary using the features provided by the Challenge Experiment 1 (Snoek et al., 2006) using the mm.dev collection and then perform the evaluation on the mm.test collection. As we are only interested in the influence of the detector performance on the search performance we only use PRFUBE with oracle weights and 10 concepts. We estimate the model parameters from the confidence scores of the real detectors and set the mean and deviation of the positive and the negative class individually. We calculate the mean and the deviation for the class $x \in \{0, 1\}$ and concept c by a common estimation method (Ross, 2006):

$$\mu_{xc} = \frac{\sum_{i=1}^{N_{xc}} o_{ic}}{N_{xc}}, \quad var_{xc} = \frac{\sum_{i=1}^{N_{xc}} (o_{ic} - \mu_{xc})^2}{N_{xc} - 1}, \quad \sigma_{xc} = \sqrt{var_{xc}}$$

Here, N_{xc} is the number of samples of the class x and o_{ic} is the observed confidence score of shot i and concept c. We perform NR=30 simulation runs. The results of the coherence study are shown in Table 7.3. We see that the average simulated detector performance of 0.13 MAP is lower than the one of the real detectors with 0.15 MAP. However, the maximal performance achieved by the simulation – among the 30 repetitions – exceeds the performance of the real detector, achieving 0.16 MAP. A possible explanation of the lower simulation performance would be the correlation of confidence scores among many shots ($\approx 2,000$) in the mm.test collection because they were near duplicates. As the proposed probabilistic detector model generates the confidence scores independently, the simulation is not able to capture these dependencies. However, we argue that the inclusion of the correlation of confidence scores in the probabilistic model is not desirable either as duplicates can be handled separately and will not be as frequent in other collections. The search performance of our model is also lower compared to the real detectors, which means that the confidence scores of used concepts were higher in the real detector set. However, three simulation runs achieve a search performance equal or higher to the real detectors. We conclude that the proposed probabilistic detector model is sufficiently realistic to explain a current, realistic retrieval setting, except the handling of near duplicates.

7.4.4 Change of Mean

Oracle Weights Figure 7.4 (a) shows the results of the simulation which increases the mean of the positive class using the MediaMill vocabulary.



(a) MediaMill vocabulary

(b) Vireo vocabulary

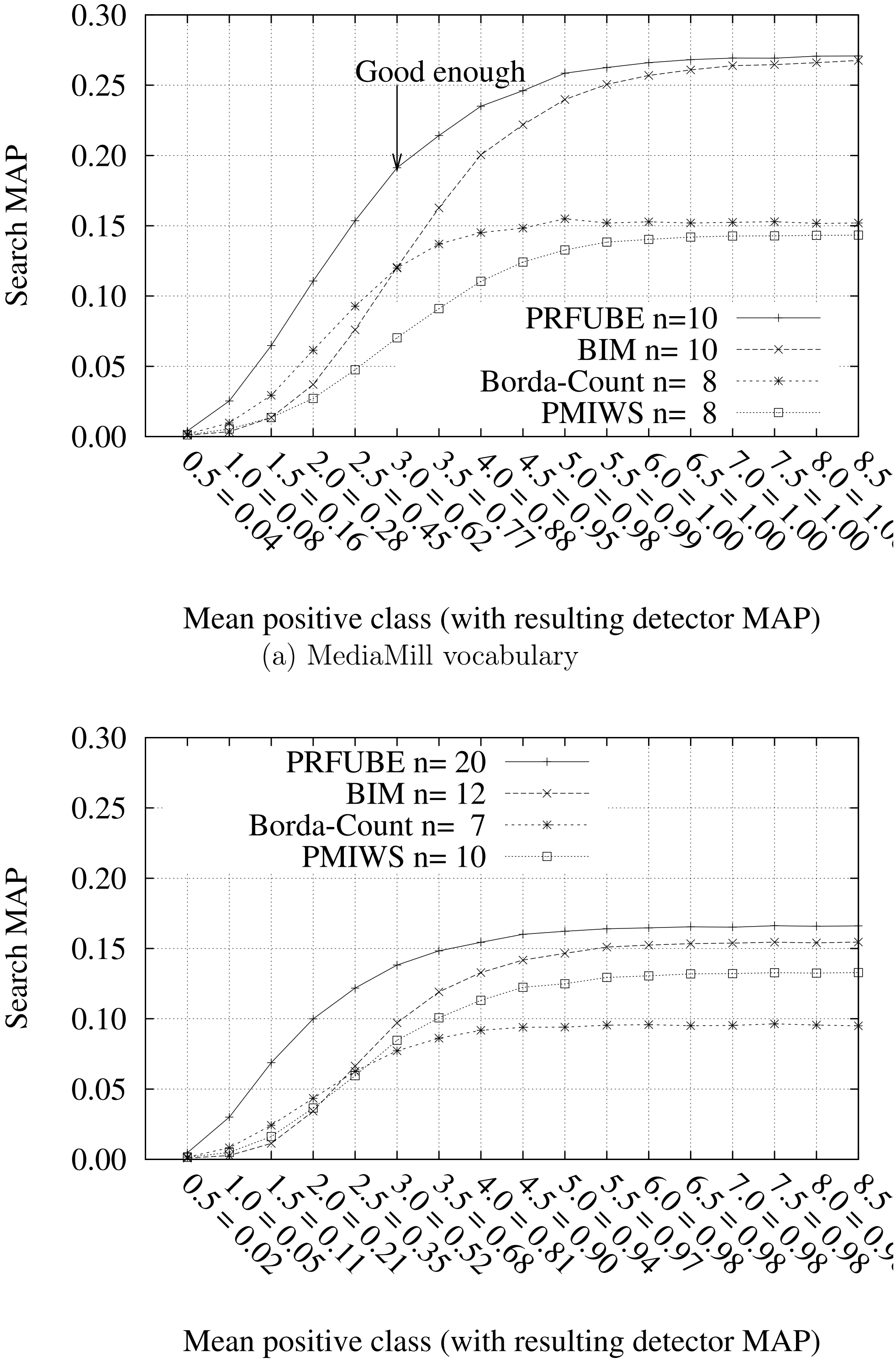
Figure 7.4: *Change of mean of the positive class μ_1 using oracle weights ($\mu_0 = 0.0, \sigma_0 = 1.0, \sigma_1 = 1.0$).*

The y-axis shows in all following figures the achieved search MAP of the depicted retrieval models. The x-axis shows the mean μ_1 together with the detector MAP which resulted from this setting, the remaining parameters are kept constant, see Figure 7.4. An increasing μ_1 leads to an increase of the detector performance. The performance of all concept-based retrieval models increases with a growing detector performance. From a positive mean of μ_1=8.5 onwards the detectors can be considered perfect classifiers. With ten concepts the PMIWS model reaches its best search performance of 0.15 MAP. Borda-Count also performs best when limited to the ten most influential concepts. It achieves an optimal performance of 0.27 MAP. The BIM method has a slow start and only reaches a search performance of 0.05 MAP at μ_1=2 which corresponds to a detector performance of 0.29 MAP. Afterwards, its performance increases faster than the two previously mentioned models and reaches at μ_1=8.5 a performance of 0.33 MAP. PRFUBE consistently shows a better search performance than all other retrieval models and achieves at μ_1=8.5 a search performance of 0.35 MAP. The BIM and PRFUBE retrieval models performed best with the use of all concepts in the vocabulary.

Figure 7.4 (b) shows the results of the simulation using the Vireo vocabulary and oracle weights. The results are similar to the usage of the MediaMill vocabulary. Notable is that this time the PMIWS method achieves a better search performance than Borda-Count. The reason is probably the existence of more only positive influential concepts - which can be exploited by the PMIWS method. The higher number of concepts allows PRFUBE to increase its search performance to 0.39 MAP.

Realistic Weights Figure 7.5 (a) and (b) show the search performance on the mm.dev collection when the weights are realistically estimated from the mm.test collection using the Annotation-Driven Concept Selection method, proposed in Chapter 4. Figure 7.5 (a) shows the simulation results of the search performance using the MediaMill vocabulary. As the weights are now estimated by a realistic concept selection method, the search performance is lower. An exception is the PMIWS method, which stays close to its performance with oracle settings of 0.15 MAP. The performance of the retrieval models relative to each other stays approximately the same. It is noticeable that Borda-Count is not able to leverage its performance gain compared to the PMIWS method in the oracle setting.

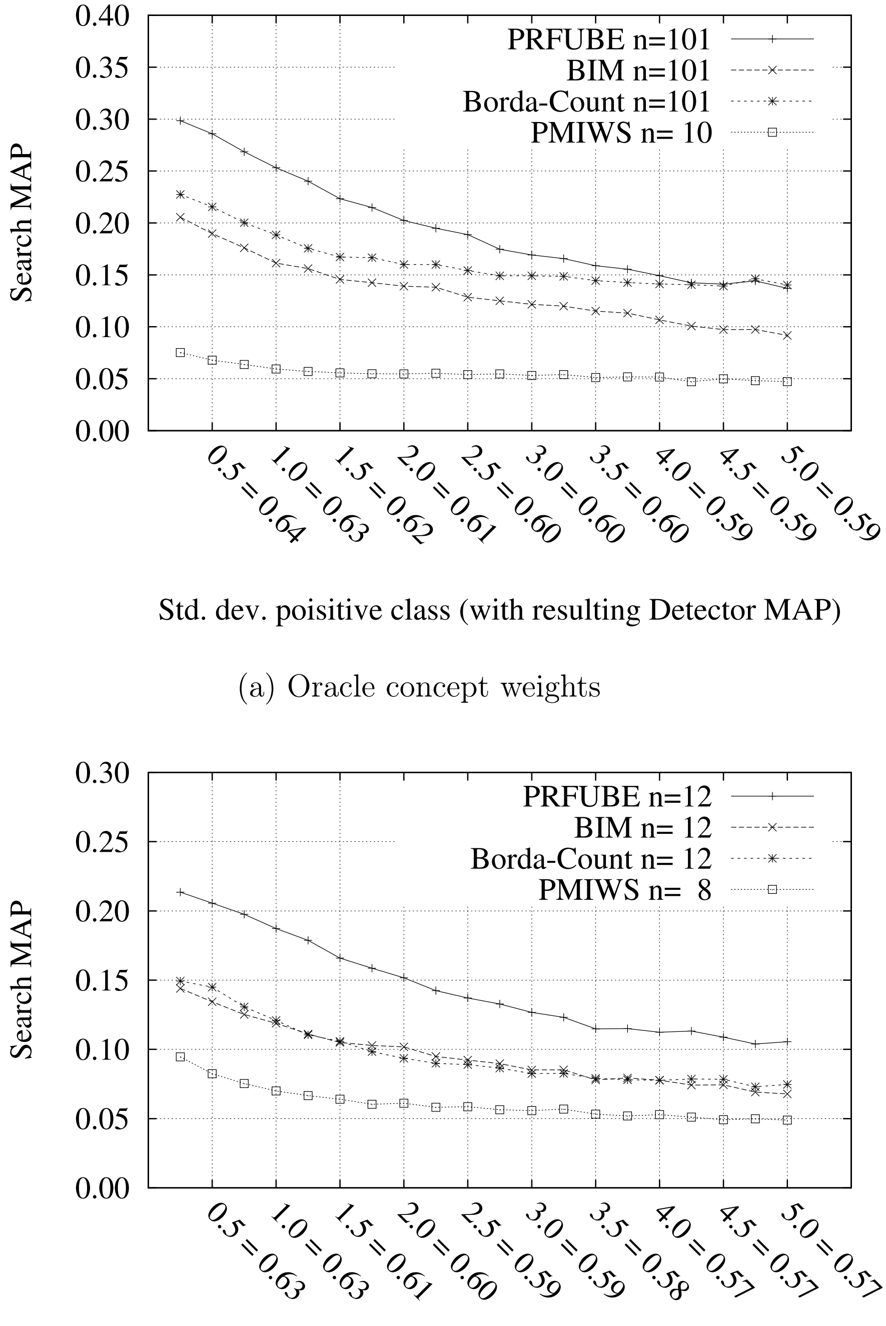
Figure 7.5 (b) shows the simulation results of the retrieval models using the Vireo vocabulary. All methods perform worse compared to the alternative of using the MediaMill vocabulary. A likely explanation is that with a growing concept vocabulary the chance of selecting poor concepts – or setting wrong weights – increases.



(a) MediaMill vocabulary

(b) Vireo vocabulary

Figure 7.5: *Change of mean of the positive class μ_1 using realistic weights (μ_0=0.0, σ_0=1.0, σ_1=1.0).*



(a) Oracle concept weights

(b) Realistic concept weights

Figure 7.6: *Change of the standard deviation of the positive class* σ_1 *using the MediaMill vocabulary* (μ_0=0.0, σ_0=1.0, μ_1=3.0)*.*

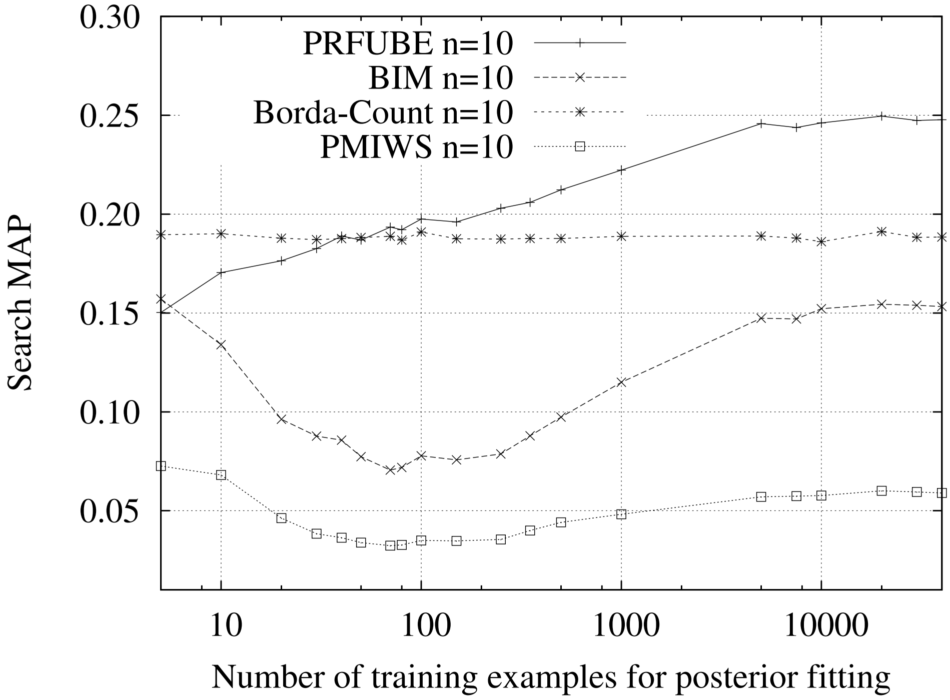


Figure 7.7: *Influence of the training examples S using the MediaMill vocabulary using oracle weights (μ_0=0.0, σ_0=1.0, μ_1=3.0, σ_1=1.0).*

7.4.5 Change of Standard Deviation

Figure 7.6 (a) shows the results of a change of the standard deviation of the positive class using oracle weight settings. We fix all other model parameters as follows: μ_0=0, μ_1=3, σ_0=1. An increase of the standard deviation of the positive class increases the uncertainty and therefore the difficulty of the search. Consequently, all retrieval models show a lower performance with an increasing standard deviation. After an initial loss of 0.07 MAP the rank-based Borda-Count performs equally with the PRFUBE from σ_1=4 onwards. The PMIWS method quickly stabilizes at MAP 0.05. The two retrieval models PRFUBE and BIM show a continuous performance loss.

Figure 7.6 (b) shows the increase of the standard deviation with weights from the realistic concept selection method. Here, PRFUBE stays around 0.03 MAP above all other retrieval models. The PMIWS method shows a worse performance than Borda-Count and BIM.

It is interesting that in both changes of the standard deviation in Figures 7.6 (a) and (b) the detector MAP only decreases by approximately 5% while the search performance reduces by approximately 50%. This suggests that the MAP performance measure of detector scores is not always a good indicator of search performance.

7.4.6 Sigmoid Fitting

Figure 7.7 shows the results of an increasing number of training examples S used in the fitting procedure for the posterior probability function. Here, we used the MediaMill vocabulary together with oracle concept weight settings. The x-axis shows the training size S on a log-scale because smaller training sizes are of higher interest. Except for small random effects, the Borda-Count method shows constant performance because it does not depend on the probabilistic output.

For the BIM and PMIWS retrieval models the search performance decreases until a number of training examples of S=100. The reason is that for a small number of training examples of S=5 the positive class was overrepresented due to the minimum number of one positive example. Therefore, the posterior probabilities are strongly biased towards higher values and the posterior probabilities and the positive classifications rise. Because the false negatives are the biggest problem for the BIM method its performance decreases. The same holds for the PMIWS method because the ranking formula only considers the probability of concept occurrence in relevant shots, see Section 5.3. With an increasing number of $S > 100$ training examples, this effect diminishes. The performance of the BIM and PMIWS models stabilizes after S=$5,000$ because of increasingly accurate estimates of the parameters for the sigmoid function.

The PRFUBE improves its search performance linearly from 0.15 MAP using 5 training examples to 0.24 MAP with $5,000$ examples. Beyond $5,000$ exsamples it stays approximately constant. It is the positively affected by over-estimated posterior probabilities of a small training example size.

7.4.7 Simulation of Video Segment Retrieval

Figure 7.8 shows the simulation results for news item retrieval described in Chapter 6. Here, we focus ourselves on changing the mean of the positive class. The results of the experiment look similar to the change of the mean for shot retrieval, see Figure 7.4. The UCLM framework performs from a detector performance of 0.16 MAP onwards better than the other retrieval models. UCLM achieves a maximal search performance of 0.384 MAP. The ECFLM method performs second best with a maximal search performance of 0.370 MAP. The Best-1 method has a slow start but then approaches the maximal performance of the ECFLM method. The CombMNZ method slowly increases its search performance to a maximal search performance of 0.186 MAP. The Borda-Count method performs the worst and achieves a maximal performance of 0.125 MAP.

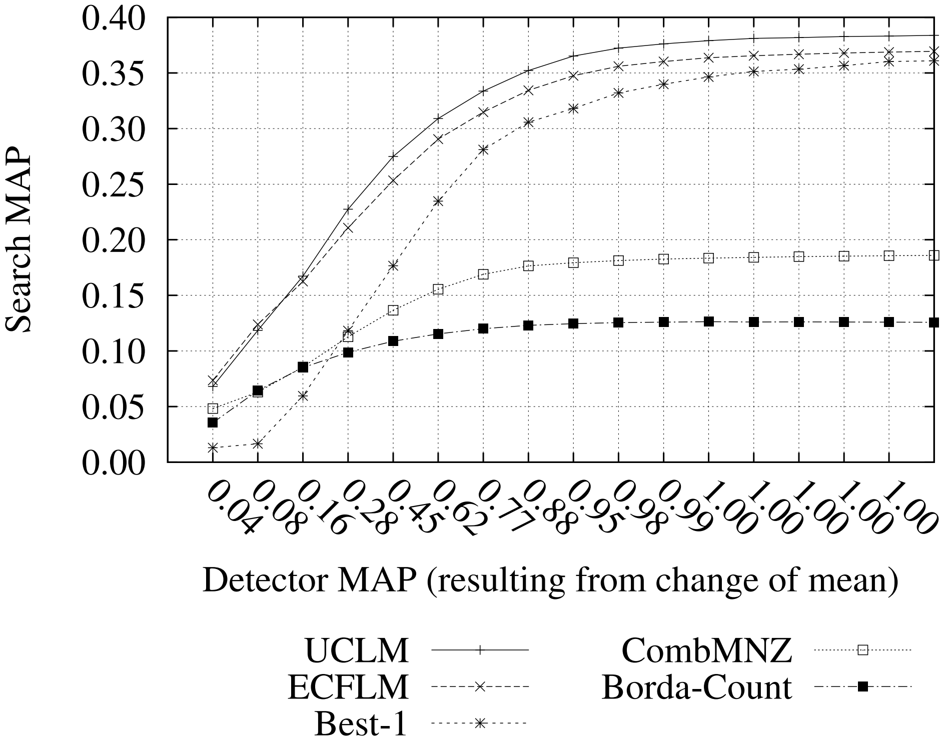


Figure 7.8: *Video segment retrieval simulation. Using realistic weights, changing the mean of the positive class* μ_1 *(*μ_0=0.0, σ_0=1.0, σ_1=1.0*).*

7.5 Summary and Discussion

This chapter proposed a Monte Carlo Simulation approach to answer the research question Q5, *How can we predict whether improved concept detection will make a current concept-based retrieval engine applicable to real-life applications in the future?* For the prediction we considered the mean average precision (MAP) of the search as a performance measure. We assume that a search performance of 0.20 MAP for a concept-based retrieval engine is sufficient for real-life applications, which is a search performance often achieved by internet retrieval engines of participants of the TREC workshop (Hawking, 2000).

The Monte Carlo Simulation then required a probabilistic model of the distribution of confidence scores of concept detectors. Here, a probabilistic model was proposed that consisted of the parameters of two Gaussian distributions, a mean and a standard deviation for each, one for concept occurred (the positive class) and one for concept absence (the negative class). The use of this model was supported by empirical evidence of real detectors and related work by Hastie and Tibshirani (1996). The Monte Carlo Simulation approach was then used to calculate the expected search performance of a retrieval engine (measured in MAP) and detector performance (also measured in MAP) given the distributions of the probabilistic model. Subsequently, we stepwise modified the parameters of the probabilistic models to improve the detector performance and noted the expected search performance of a set of retrieval models. This allowed us to predict the expected search performance of retrieval engines when the detector performance improves.

The experiments were carried out on the TRECVid 2005 development collection, where relevance and concept occurrences were known. We used two concept vocabularies: the MediaMill vocabulary consisting of 101 concepts and the Vireo vocabulary consisting of 374 concepts. Furthermore, we investigated the influence of the concept selection and weighting method by considering two methods. First, using an oracle concept selection and weighting method, which selected the concepts and their weights in hindsight with knowledge of relevance. Second, the realistic Annotation-Driven Concept Selection method, proposed in Chapter 4.

Video Shot Retrieval For video shot retrieval, the development of the search performance with increasing detector performance of four retrieval models was simulated. When increasing the mean of the positive class, we found that the two retrieval models based on concept-based document representations, the Binary Independence Model (BIM) and the Probabilistic Framework for Unobservable Events (PRFUBE), can exploit the full concept vocabulary under an oracle weight setting. However, the search performance of BIM increases slower due to a high misclassification rate with low detector performance. Borda-Count first showed similar performance to BIM but reached a lower search MAP. The Pointwise Mutual Information Weighting Scheme (PMIWS) method has a lower performance than the other methods. The maximum reached search performance is 0.39 MAP by the PRFUBE and using the Vireo vocabulary. Additionally, in most experiments PRFUBE showed the best performance among the four retrieval models. PRFUBE is the first to achieve real-life sufficient performance under realistic weight settings with an approximate detector performance of 0.60 MAP - which is still far from perfect classification. The only other retrieval model which achieved a search performance of 0.20 MAP was BIM but at a much higher detector performance of 0.88 MAP. We therefore predict that retrieval models using concept occurrence based document representation generally perform better than retrieval models based on confidence scores and they will be applicable to real-life applications once concept detectors reach a high performance level of 0.60 MAP or above.

We also found that the MAP performance measure for concept detectors is not necessarily a good indicator of the search performance since the increase of the standard deviation of the positive class caused a severe search performance decrease while the detector performance reduced only slightly. We plan to investigate other measures which consider the overlaps of the confidence score distributions from the positive and the negative class, such as the Kullback Leibner Divergence (Arndt, 2001), in future work.

Furthermore, we investigated the influence of fitting errors of the posterior probability function. While Borda-Count was unaffected - because it only depends on confidence scores not on a probability measure – all other retrieval models showed decreased performance beneath $5,000$ training examples.

Video Segment Retrieval For video segment search we focused on the simulation of increasing the mean of the positive class, which increases the detector performance. Here, the Uncertain Concept Occurrence Language Model (UCLM) and the Expected Concept Frequency Language Model (ECFLM) achieved a real-life applicable search performance of 0.20 MAP at a detector performance of 0.28 MAP. From this performance, the UCLM method performed better than the ECFLM model, reaching a maximal performance of 0.384 MAP and 0.370 MAP respectively. The Best-1 model, which used concept-occurrence classifications, achieved a real-life applicable performance at a detector performance of approximately 0.50 MAP. The two retrieval models which are based on confidence scores, CombMNZ, see Equation 2.4.2 and Borda-Count did not achieve real-life applicable performances.

Chapter 8

Conclusions and Future Work

This thesis investigated the merits of *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval.* In Chapter 1 five research questions (Q1-Q5) were derived from important problems in concept-based retrieval. In this chapter, Section 8.1 draws conclusions from the answers to these questions given in this thesis. Section 8.2 proposes future work. Finally, Section 8.3 ends this thesis with some concluding remarks.

8.1 Conclusions

In this section, conclusions are drawn from the proposed answers in this thesis to the research questions enumerated in Section 1.5, which will be repeated in the text below.

8.1.1 Uncertain Representations Ranking Framework

This thesis found the following answers to research question Q1, *Can a general framework be defined for document representation uncertainty, which re-uses text retrieval for concept-based retrieval?* In Chapter 3 the Uncertain Representation Ranking (URR) framework was proposed which ranks documents with uncertain document representation re-using a text retrieval ranking function for concept-based retrieval. The URR framework was inspired by the principles of the Portfolio Selection Theory from Markowitz (1952) which optimizes the percentages of the available budget which should be invested in a particular share, called the investment percentages.

Although treating different problems, the following parallels between the proposed URR framework and the Portfolio Selection Theory were identified:

- Documents in a collection can be seen as shares in a stock market (a collection of shares).
- The score under which a document would be ranked by a text score function, if the document representation was known, corresponds to the future win of a share.

- The problem of ranking documents by the uncertain score corresponds to the list of shares sorted by their investment percentages.

In the Portfolio Selection Theory, the investment percentages are chosen in such a way that a mixture of the expected total win and its variance are optimized. The influence of the variance to this mixture is regulated by a risk parameter, which represents the risk attitude of an investor. The expected total win and its variance can be computed by three kinds of components which are assumed to be known: the expected win of each share, their variances and the co-variances between the shares. Now, instead of optimizing the investment percentages, the URR framework optimizes the ranking of documents.

In parallel to the Portfolio Selection Theory, the URR framework considers the expected score and its variance. Note, since no probabilistic model for the co-variances of document scores existed, the co-variance of the score was left for future work. However, there is a difference between how the Portfolio Selection Theory and the URR framework model the expected wins/scores and their variance. In the Portfolio Selection Theory the expected win and its variance are assumed to be *known*, while in the URR framework the reason for the uncertainty of the score is explicitly modeled. The URR framework assumes that the text score function returns the score based on the document's representation but the document's representation is uncertain and this causes the uncertainty of the score. As a result, the expected score and its variance were calculated using the probability distribution of concept occurrences defined during the concept detection.

Similar to the Mean-Variance Analysis framework, which has been recently proposed by Wang (2009) for uncertain text scores, the URR framework ranks documents with uncertain document representation under a mixture of the expected score and the score's standard deviation. The mixture is controlled by a risk parameter, which specifies the influence of the standard deviation (the uncertainty) on the final ranking score value. The standard deviation, being the square root of the variance, was employed instead of the variance because the risk parameter was easier to set this way. The applicability of the URR framework to concept-based retrieval was shown in experiments in Chapter 5 and Chapter 6, where the framework was applied using two different text retrieval functions. This leads to the following conclusion.

Conclusion 1. *The document representation uncertainty framework provides a general and theoretically founded way to re-use text retrieval functions in concept-based retrieval. Documents are ranked by their expected score and its standard deviation, resulting from the multiple possible document representations of a document.*

8.1.2 Automatic Concept Selection and Weighting

Chapter 4 provided answers to the research question Q2, *How can the document representation and its weights be defined automatically and in a user-friendly manner for an information need?* The desirable properties of such methods were identified to be the following:

- The retrieval engine should maximize the user's freedom to formulate his query. For example, the user should be allowed to use text queries, as in internet retrieval engines, without knowledge of the concept vocabulary used by the retrieval engine.

- The selection of concepts and their weights should be collection-specific. For example, for a general video collection and the information need "President Obama" the concept *U.S. Flag* should probably be selected for searching with a high weight. On the other hand, in a collection of documentaries on the U.S. presidents, this concept would only introduce noise and should therefore be ignored, since most U.S. presidents will be shown with a U.S. Flag.

In order to fulfill these requirements, the Annotation-Driven Concept Selection (ADCS) method was proposed. It builds a textual representation of the development collection, which was fully annotated with the concepts of a concept vocabulary. Such development collections, are normally created for training concept detectors. For a textual query, a standard text retrieval engine was used to produce a ranked list of shots. The quality of the ranked list was of a performance similar to current internet retrieval engines. The text retrieval ranking was then used together with the known concept occurrences in the shots to estimate the occurrence probability of concepts in relevant shots. This probability is used as a weight in several retrieval functions. Furthermore, the Mutual Information was calculated between a concept and relevance which in turn was used to select good concepts for the current information need. The ADCS method allows the user to formulate its query in text, which many users are used to today. The method does not require knowledge of the concept vocabulary nor of the collection from the user. Therefore, the method is likely to be a good support for users to formulate their queries.

It was shown through experiments that the proposed concept selection method performs better in terms of the mean average precision and the rank correlation compared to the text matching baseline proposed by Huurnink et al. (2008) and the wiki-article baseline which was proposed by us, see Hauff et al. (2007). Additionally, in Chapter 5 we showed that the concept selection method achieves significant performance improvements. This leads to the following conclusion.

Conclusion 2. *Using textual representations of collections with known concept occurrences outperforms the text matching baseline proposed in the literature, in terms of mean average precision and rank correlation. The provided* *weights can be used in several existing retrieval functions. The concept selection method is likely to support users in formulating their query since no knowledge of the concept vocabulary or the collection is required and the query can be specified in text, what many users are used to today.*

8.1.3 Support of Longer Video Segments

It is likely that users are interested in other results than whole videos or single video shots. Therefore, Chapter 6 answered the research question Q3, *How can the retrieval of longer video segments be supported based on concept occurrence in video shots?* This question has been raised occasionally, but up till now was under addressed. Therefore, our approach was limited to the retrieval of news items in broadcast news videos, which is an example of a longer video segment.

An existing segmentation algorithm for broadcast news videos into news items was used and the news items were modeled as a series of shots each with binary concept occurrences. As a result, the concept-based document representations consisted of concept frequencies (counting the occurrences) rather than single concept occurrences which are normally considered in shot retrieval. This representation is similar to document representations of term frequencies in text retrieval. Here, the term frequency is used as a measure of the importance of a term in a document. Intuitively, this is also true for the concept frequency. While retrieval performance of this approach is discussed as an answer to research question Q4, this leads to the following conclusion.

Conclusion 3. *Because of the similarity of term frequencies in text retrieval, concept frequencies are a good measure of the importance of a concept for a document and therefore suitable as a document representation of longer video segments.*

8.1.4 Performance Impact of the URR Framework

The answers to research question Q4, *What is the impact of the proposed ranking framework and the concept selection and weighting method on the retrieval performance?*, were given in Chapter 5 and Chapter 6 which applied the URR framework to different document representations. Both instances are described in the following.

Shot Retrieval In Chapter 5 the Probabilistic Framework for Unobservable Binary Events (PRFUBE) was proposed. Here, the probability of relevance (text) ranking function (Robertson, 1977) was re-used together with the URR framework to rank video shots with unknown concept occurrences. Two variants of the application of the URR framework were investigated:

(1) A direct application of the URR framework, which calculated the expected score and the variance from the 2^n possible document representations when considering n concepts.

(2) An operational ranking function having a linear runtime complexity which made the assumption that the prior probabilities of the used concepts were independent. This variant is referred to as the PRFUBE framework.

The proposed concept selection and weighting algorithm improved the performance of the PRFUBE framework compared over a baseline approach using text retrieval scores on concept descriptions. Furthermore, an overall comparison between seven representative retrieval functions using six pairs of collections and detector sets including different video domains was undertaken. The performance of the PRFUBE framework was always among the best two engines. Where it was worse than another engine, this difference was not significant according to a Wilcoxon signed-rank test with a significance level of 0.05. Finally, retrospective experiments showed that the method further benefits from better selection and weightings. This leads to the following conclusion.

Conclusion 4. *The use of the PRFUBE framework, an application of the URR ranking framework to the probability of relevance ranking function with binary document representations for shot retrieval, improves the retrieval performance in many collections and shows better average performance than other retrieval functions.*

Segment Retrieval In Chapter 6 the language modeling ranking function was used together with the URR framework to rank news items in the uncertain concept (occurrence) language modeling (UCLM) method. Here, concept frequencies were used as a document representation for a news item. The Monte Carlo Simulation approach (Metropolis and Ulam, 1949) was used to approximate the expected score and the standard deviation. The UCLM was evaluated using relevance judgments derived from a shot retrieval task. Furthermore, no baselines techniques existed. In order to compare the results of the UCLM method, five baselines, representing extensions of existing techniques from shot retrieval, were defined. First, three models which originated from confidence score based models, namely CombMNZ, CombSUM and Borda-Count, see Aslam and Montague (2001). Here, the used confidence score was the average confidence score of the shots in the video segment. Second, two retrieval models using concept language models in a different way were investigated. First, the Best-1 method used the classification output of the concept detectors to estimate the concept frequency. Second, the expected concept frequency language model used the expected concept frequency instead of the actual concept frequencies in the language model framework. The achieved performance of the UCLM approach was significantly better than all five baselines. Furthermore, including the standard deviation of the score with a positive risk attitude resulted in significant improvements over only using the expected score. This leads to the following conclusion.

Conclusion 5. *The use of the UCLM method, an application of the URR framework, which re-uses the language modeling ranking function, improves retrieval performance for longer video segments.*

8.1.5 Simulation of Improved Detector Performance

Current concept detector performance is still low. This strongly influences the retrieval performance. Chapter 7 answered the research question Q5, *How can we predict whether improved concept detection will make a current concept-based retrieval engine applicable to real-life applications in the future?* The answer was given by using a Monte Carlo Simulation approach to simulate detectors with increasing performance to study their influence on several state-of-the-art retrieval models.

The output from the proposed simulation approach was the detectors' confidence scores, which were then translated into the posterior probability of concept occurrences and binary classification output. Monte Carlo Simulations require a probabilistic model for the inputs of the simulation, the confidence scores of the detectors. The proposed probabilistic model consisted of the parameters of two Gaussian distributions (a mean and a variance) one for shots where the concept occurred (the positive class) and one for shots where the concept was absent (the negative class). For a given parameter setting of the probabilistic model and a collection with known concept occurrences, a simulation run was performed as follows. The simulation result was the expected search performance of a retrieval model for a given expected detector performance. The advantages of this model are that the simulation parameters and the detector performance can be adjusted stepwise. We assume that a search performance of 0.20 MAP for a concept-based retrieval engine is sufficient for real-life applications, which is a search performance often achieved by internet retrieval engines of participants of the TREC workshop (Hawking, 2000).

The first experiment stepwise increased the mean of the Gaussian distribution for the positive class and investigated the resulting performance of video shot retrieval. The two retrieval models which were based on (the rank of) confidence scores showed a modest increase of the search performance with an increasing detector performance and stabilized at a low search performance compared to the other two models. The retrieval model which was based on classification output showed a slower increase in search performance but stabilized at a higher search performance, given nearly perfect detectors. For the PFUBE framework, which was proposed in Chapter 5, the search performance increased faster with the detector performance than with other retrieval models and stabilized at the same high performance as the retrieval model which was using the classification output. The search performance threshold of 0.20 MAP, which was assumed to be sufficient for real-life applications, was achieved by the PRFUBE framework at the lowest detector performance level of 0.62 MAP. Given the low performance of today's detectors, for a retrieval engine to achieve this performance, concept detectors still need to improve, but then concept-based retrieval is a promising approach to multimedia retrieval. This leads to the following conclusion.

Conclusion 6. *Under improved concept detector performance, the search performance of the PRFUBE increased the fastest and achieved the highest performance, outperforming confidence score and classification based retrieval models. A sufficient search performance level of* 0.20 *MAP (see above) was achieved by the PRFUBE framework at a detector performance of* 0.62 *MAP. Therefore, once concept detectors achieve this performance, concept-based multimedia retrieval will be ready for real-life applications.*

The second experiment simulated detectors which had more extreme differences in their confidence about concept occurrences. This was achieved by fixing the mean of the positive class at a high value and stepwise increasing the variance of the positive class. As a result, the detector and search performance of the described four retrieval models decreased. Furthermore, the distinct decrease of the search performance of 0.15 MAP was disproportionate to the slight decrease of the detector performance of 0.03 MAP. Using perfect concept selections and weightings, the Borda-Count retrieval model, which only takes into account the ranks of the confidence scores (rather than their values) showed the most stable performance. However, using a realistic concept selection and weighting algorithm, the PRFUBE framework stayed at a higher performance than four other retrieval models. This leads to the following conclusion.

Conclusion 7. *The search performance of today's retrieval models is sensitive to concept detectors with a high confidence score variance.*

Additionally, the disproportionate decrease of the search performance compared to the detector performance is an unexpected finding of this thesis. Although the precise reason for this was not investigated, the finding indicates that the MAP measure for detector performance is not always a good indicator for the usefulness of a set of detectors for retrieval. This leads to the following conclusion.

Conclusion 8. *The mean average precision performance measure of a detector set does not fully explain the helpfulness of a detector set for retrieval.*

The final experiment stepwise increased the mean of the Gaussian distribution for the positive class and investigated the resulting performance of video segment retrieval. Here, the impact of the changed detector performance on the UCLM framework, proposed in Chapter 6, and five proposed baselines, see Section 8.1.4, was investigated. The outcome was similar to the change of the detector mean for the positive class. The CombMNZ method, as the best confidence score based retrieval model, achieved a maximum search performance of 0.18 MAP. On the other hand, the methods based on the confidence occurrence achieved a performance above 0.35 MAP and increased faster in search performance given an increase of the detector performance. The UCLM approach achieved the best performance with 0.38 MAP under perfect detection. This leads to the following conclusion.

Conclusion 9. *Concept-based retrieval models which are based on concept frequencies reach sufficient search performance under a lower detector performance.*

Furthermore, the UCLM framework and the Expected Concept Frequency Language Model (ECFLM) achieved a real-life applicable search performance of 0.20 MAP at a detector performance of 0.28 MAP, which is a much lower performance than required for shot retrieval. Since a state-of-the-art detector set achieves a detector performance of up to 0.40 MAP (van de Sande et al., 2010) this suggests that a real-life applicable search performance is possible today for the particular collection. This leads to the following conclusion.

Conclusion 10. *At least for news item search under the given setting, video segment retrieval is an easier task for concept-based retrieval engines than video shot retrieval. Since one can argue that the tasks are equally relevant to real-life applications, video segment retrieval is an important research direction to pursue.*

8.2 Proposed Future Research

In the course of the work reported here, investigations were sometimes limited to basic approaches in order to be able to focus on the most important aspects of the topic. Furthermore, for some findings in this thesis, it is worthwhile to investigate, whether they can also be applied to other areas in information retrieval. Therefore, a description of these investigations are proposed for future research and listed in the following.

8.2.1 Uncertainty Modeling in Information Retrieval

Document Dependencies Inspired by the co-variance of share wins in the Portfolio Selection Theory and the nature of videos, it seems likely that the representation of some video segments will depend on each other, for example if one is situated right after the other. However, as yet there are no probabilistic models available for these dependencies which would allow us to exploit them. Therefore, the exploration of document dependency models is proposed for future work.

Integration and Comparison of Uncertainties Models There is work on other kinds of uncertainties in an information retrieval engine than the URR framework: Wang (2009) proposes in the Mean Variance Analysis framework to consider the score of a document to be distributed around the result of the score function. Furthermore, the query classes by Yan (2006) assume that there are multiple score functions, each having different weights, and it is unknown which score function is the correct one for a given information need. The two above approaches are orthogonal to the URR framework and it is proposed for future work, to investigate whether they can be integrated.

Adaptation to Spoken Document Retrieval Although the spoken terms in recordings are different features than concept occurrence, they carry the same characteristics: the automatic speech recognition engines also make observations, which correspond to the confidence scores of concept detectors, and the decision as to which terms have been said is commonly made based on probabilistic models. Therefore, the URR framework could also be applied to spoken document retrieval.

Include Detector Performance Throughout this thesis, all concept detectors were treated equally. However, it is a known fact that the detector performance differs between frequent and rare concepts. Furthermore, Yang and Hauptmann (2008b) find that concept detector quality strongly depends on the domain the detectors are trained on. Therefore, it is proposed to investigate how these performance differences can be integrated.

Improved Sampling Methods The number of possible representations of a document quickly grows too large to consider all of them when determining the expected score and its variance. In this thesis, the Monte Carlo Simulation approach was proposed to reduce the number of considered document representations. However, there are more advanced sampling methods, which further reduce the required number of samples. Therefore, the integration of these methods is proposed for future work.

8.2.2 Concept Selection and Weighting

Evaluation Concept Weightings In this thesis only the concept selection part of a concept selection and weighting algorithm are evaluated. However, the weights assigned to each of these concepts also influence the final retrieval result. Therefore, future work should investigate how to evaluate the assigned values of concept weights.

Inclusion of Search Collection The proposed concept selection method operated on an annotated development collection and it was assumed that concept occurrences in relevant shots were similarly distributed in the search collection. However, for collections from different domains this might not hold. Therefore, future work should investigate ways to integrate the conditions of the concept occurrences in the search collection with the proposed concept selection and weighting algorithm.

8.2.3 Retrieval of Longer Video Segments

Concept Based Segmentation Besides the retrieval of longer segments, it is also difficult to first segment a video into semantic units. Until now, successful segmentation was only achieved with video data which had some regularity. For example, an anchorman who appears at the beginning of each news item can be used for segmentation. However, with more diverse data it is difficult to recognize patterns in the video. On the other hand, if a video has been described by a series of shots with actual concept occurrences and absences, it is possible that the concept occurrences have a clearer pattern. For example, if a sequence of shots contains *People* and *Animals*, surrounded by shots which contains *Streets* and *Houses* this pattern could be used to create three segments, assuming that for each shot boundary a segment boundary score could be calculated. However, the concept occurrences are unknown and using the parallel to the URR framework the expected segment boundary score can be calculated. The video should then be segmented by the highest expected segment boundary score in a region of shots.

Task Evaluation Evaluation platforms, such as TRECVid and TREC, are already major enablers of many information retrieval tasks due to the provision of standardized collections, relevance judgments and evaluation methodology. Therefore, future work should investigate whether the retrieval of longer video segments could be included as a new search task, to make research in this direction comparable.

8.2.4 Detector Simulation

Different Simulation Parameters In the proposed simulation all concepts used the same simulation parameters except of a different prior. However, it is realistic to assume that detectors for frequent concepts are more accurate (have a better parameter setting) than rare concepts. Furthermore, the assumption that either the mean of a class or its variance changes is unrealistic, since a better detector model (higher difference in the class's mean) is supposedly less confident at more seldom shots (which increases the variance). Therefore, future research should investigate how the simulation parameters can be varied in a more realistic way.

Evaluation of detectors The increase of the variance of the positive class resulted in a disproportionately high decrease of search performance compared to the detector performance, both measured by the mean average precision. This suggests that the mean average precision of the detector output is not always a good indicator of the search performance which is the ultimate goal of a retrieval engine. Therefore, the reasons for the disproportion between the search performance measures and possible better measures of detector performance should be researched in the future.

8.3 Concluding Remarks

Concept-based multimedia retrieval has a great potential to allow users to search in multimedia collections. Since it is independent from costly and language-dependent textual metadata, the concept-based retrieval paradigm allows searching in different modalities, which is problematic for other retrieval methods. This thesis has shown principal ways to help achieving this goal. Although concept detector research is still endeavoring to improve detector performance, simulations performed in this thesis suggest that once this has been achieved concept-based multimedia retrieval will be applicable to large-scale real world applications in the future.

Appendix A

Extract from Probability Theory

This appendix is an elaboration of the introduction of probabilities in Section 2.2.1. The elements from this theory are defined here.

Basic Definitions

Events, Event Space and Probability Measure Probabilities are always used in reference to a probabilistic event space (a set of events). Throughout this thesis the uniform probability measure will be assumed, where each event has the same probability.

Random Variables A random variable Z is a function of an event to the function's range. As common in probabilistic notations, random variables will be denoted in upper case. Note, that the definition of a feature and random variable are similar. However, to conform to standard notation, random variables will be used in the context where a set of documents is considered, while the feature notation will be used if the focus is a single document. A random variable is called discrete, if the range is a countable set $dom(Z)$. Note, most often used ranges are subsets of the natural numbers, $dom(Z) \subseteq \mathbb{N}$. However, in this thesis also other sets are used, which is why the definition is kept more general.

Prior and Conditional Probability Now, two probabilistic functions, which are used in this thesis, are defined. Let $\mathcal{Z}$ be a probabilistic event space and $P_{\mathcal{Z}}$ be the probability measure on this space. Furthermore, let Z_1 and Z_2 be two discrete random variables. The prior of Z_1 and the conditional probability of Z_1 given a value of Z_2 are defined as follows; for $k \in dom(Z_1), l \in dom(Z_2)$:

$$P_{\mathcal{Z}}(Z_1 = k) = \frac{|\{e \in \mathcal{Z} | Z_1(e) = k\}|}{|\mathcal{Z}|}$$

$$P_{\mathcal{Z}}(Z_1 = k | Z_2 = l) = \frac{|\{e \in \mathcal{Z} | Z_1(e) = k, Z_2(e) = l\}|}{|\{e' \in \mathcal{Z} | Z_2(e') = l\}|}$$

The prior and conditional probability of vectors of random variables, $\vec{Z}_1$ or $\vec{Z}_2$, are equivalently defined to the one above. For a Boolean random variable $Z_1 \in \mathbb{B}$ the common notation of Z_1 for $Z_1 = 1$ and $\bar{Z}_1$ for $Z_1 = 0$ is used. Often, the name of the random variable will be left out for brevity reasons if the value indicates which variable it belongs to, for example: $P(Z_1|z_2)$ is the probability of Z_1 being 1 given that the random variable Z_2 is equal to z_2.

Models for Probabilities Probabilities are seldom known exactly and there two common ways to model them: Generative models and discriminative models. Let Z_1 be a discrete and Z_2 a continuous random variable. In a generative model using the Bayesian view of probabilities a probability is given as follows; for $k \in dom(Z_1)$ and $l \in \mathbb{R}$:

$$P_{\mathcal{Z}}(Z_1 = k|Z_2 = l) = \frac{p(Z_2 = l|Z_1 = k)P_{\mathcal{Z}}(Z_1 = k)}{\sum_{k' \in \mathbb{N}} p(Z_2 = l|Z_1 = k')P_{\mathcal{Z}}(Z_1 = k')} \tag{A.1}$$

Here, $p(Z_2 = l|Z_1 = k) \in \mathbb{R}^+$ is the likelihood of Z_2 being l, given Z_1 is k. On the other hand, a discriminative model defines the posterior probability through a function which maps its input variables directly (without the detour of likelihoods and priors) to the interval $[0:1]$. For example, for $k \in dom(Z_1)$ and $l \in \mathbb{R}$:

$$P_{\mathcal{Z}}(Z_1|Z_2 = l) = \frac{1}{1 + exp(-l)} \tag{A.2}$$

Here, the right side is a logistic function mapping the values of Z_2 to the interval $[0:1]$.

Functions of Random Variables The notion of a function of random variables is seldom used in information retrieval. If Z is a random variable defined on an event space $\mathcal{Z}$ and $f : dom(Z) \to \mathcal{X}$ is a function mapping values from the domain of Z to another set $\mathcal{X}$. We will write $F := f(Z)$ and mean another random variable F which is defined as:

$$F(e) = f(Z(e)), \quad \forall e \in \mathcal{Z}$$

Here, the random variable F maps each event e to the application of f on the result of the application of Z on the event e. We will use the notation as a function (f instead of F) for the calculation of expected values and variances.

Expectations of Random Variables We now define the expected value of a function of a random variable given the value of another random variable. Let $Z_2 := f(Z_1)$ be a function of the random variable Z_1. The expected value of Z_2 is defined as follows.

$$E[Z_2] = \sum_{z_1 \in dom(Z_1)} f(z_1)P_{\mathcal{Z}}(Z_1 = z_1) \tag{A.3}$$

Furthermore, the variance of Z_2 is defined as follows:

$$\mathrm{var}[Z_2|z_3] = E[Z_2^2] - E[Z_2]^2 \tag{A.4}$$

with

$$E[Z_2^2] = \sum_{z_1 \in dom(Z_1)} f(z_1)^2 P_{\mathcal{Z}}(Z_1 = z_1) \tag{A.5}$$

The co-variance between two random variables Z_1 and Z_2 is defined as follows:

$$\mathrm{cov}[Z1, Z_2] = E[Z_1 Z_2] - E[Z_1]\ E[Z_2] \tag{A.6}$$

Additionally, following laws of expectations and variances hold.

$$\begin{aligned} E[c\ Z_1] &= c\ E[Z_1] && \text{(A.7)} \\ E\left[\sum_{i=1}^{n} Z_i\right] &= \sum_{i=1}^{n} E[Z_i] && \text{(A.8)} \\ \mathrm{var}[c\ Z_1] &= c^2\ \mathrm{var}[Z_1] && \text{(A.9)} \\ \mathrm{var}\left[\sum_{i=1}^{n} Z_i\right] &= \sum_{i=1}^{n}\left[\mathrm{var}[Z_i] + \sum_{j=1, j\neq i}^{n} \mathrm{cov}[Z_i, Z_j]\right] && \text{(A.10)} \end{aligned}$$

Here, c is a constant and Z_i with $1 \leq i \leq n$ are random variables. Instead of expectations we also sometimes use conditional expectations. In this case, the probability distribution is a conditional probability distribution. For example, let $Z_2 := f(Z_1)$ be a function of the random variable Z_1. The conditional expected value of Z_2 given a value z_3 of a random variable Z_2 is defined as follows.

$$E[Z_2|z_3] = \sum_{z_1 \in dom(Z_1)} f(z_1) P_{\mathcal{Z}}(Z_1 = z_1|z_3) \tag{A.11}$$

Here, $P_{\mathcal{Z}}(Z_1|z_3)$ is the conditional probability distribution. Since variances and co-variances can be defined as expectations, they can also be conditional, using the same notation as for expectations.

Document Specific Random Variables

An important notion in the Portfolio Selection Theory (Markowitz, 1952), the Mean-Variance Analysis framework (Wang, 2009) and in the proposed Uncertain Representation Ranking framework, proposed in Chapter 3, are random variables which are specific to a single document (or a share). This kind of random variables is rarely used in information retrieval where random variables are normally used to make statements over groups of documents.

To improve the intuition, the notion of document specific random variables is explained using the demonstrative urn metaphor from Section 2.8.

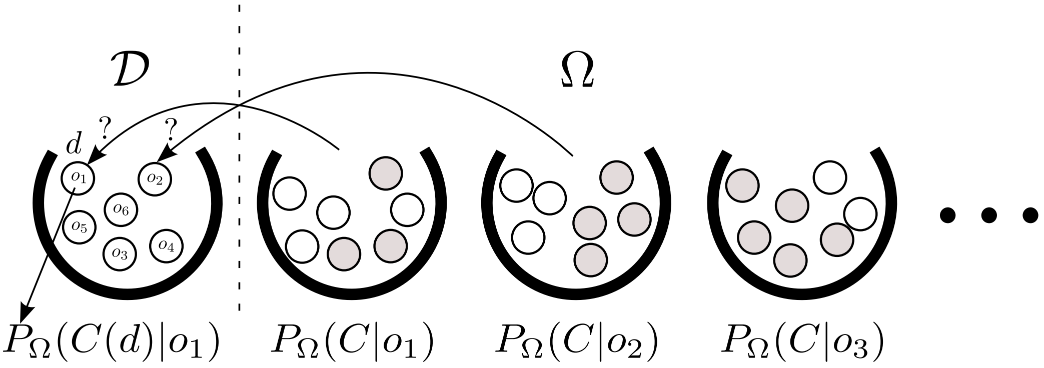


Figure A.1: *Visualization of the meaning of a document specific random variable.*

Thought Experiment A.1. *Considering a single concept C, we only know the confidence score o_1 of the detector for C for the document d_1 in the collection $\mathcal{D}$. The document d_1 can be seen as a wrapped ball which was randomly drawn from an urn labeled o_1, in which all ball in the universe of document Ω are placed which have this confidence score. The probability to draw* any *document in which a concept occurs with confidence score o_1 is $P_\Omega(C|o_1)$. Therefore, if we now consider a particular document d in the collection $\mathcal{D}$, we can describe its uncertain concept occurrence by the random variable $C(d)$. Because the document was randomly drawn, the probability of the concept occuring in d is $P_\Omega(C(d)|o_1)$ which is equal to the probability of the concept occuring in* any *document with confidence score o_1, $P_\Omega(C|o_1)$.*

Furthermore, if we define a score function $score_q : \vec{F} \rightarrow \mathbb{R}$ for any *document d with a certain representation $\vec{f}(d)$, all documents in Ω carry a score value. However, since the document representation is uncertain, we describe the score value by the random variable $S(()d)$. In the above case of a document representations of only one concept , this score value can only take one of two different values, namely $score_q(0)$ and $score_q(1)$. The probability that the document d takes the value $score_q(1)$ is equal to the probability that the concept occurs occurs in the document, $P(C(d)|o_1)$. If we increase the number of considered concepts in the document representation $\vec{F}$, the number of possible score values increases.*

If longer video segments are moldeled as series of shots, a segment d can be imagined as set of balls $d.s_1, \dots, d.s_{dl}$. Considering only a single concept, we assume each was randomly drawn from the urn $o(d.s_i)$ respectively. As a result, there are $dl + 1$ possible term frequencies for the document d, each with a certain probability. If a score function is now defined on the term frequency it can also take as many different values.

Appendix B

Siks Dissertations

Since 1998, all dissertations written by Ph.D. students who have conducted their research under auspices of a senior research fellow of the SIKS research school are published in the SIKS Dissertation Series.

2010-33 Robin Aly (UT) *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*

2010-32 Marcel Hiel (UvT) *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*

2010-31 Viktor de Boer (UVA) *Ontology Enrichment from Heterogeneous Sources on the Web*

2010-30 Marieke van Erp (UvT) *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval*

2010-29 Stratos Idreos (CWI) *Database Cracking: Towards Auto-tuning Database Kernels*

2010-28 Arne Koopman (UU) *Characteristic Relational Patterns*

2010-27 Marten Voulon (UL) *Automatisch contracteren*

2010-26 Ying Zhang (CWI) *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*

2010-25 Zulfiqar Ali Memon (VU) *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*

2010-24 Dmytro Tykhonov Designing Generic and Efficient Negotiation Strategies

2010-23 Bas Steunebrink (UU) *The Logical Structure of Emotions*

2010-22 Michiel Hildebrand (CWI) *End-user Support for Access to*
Heterogeneous Linked Data

2010-21 Harold van Heerde (UT) *Privacy-aware data management by means of data degradation*

2010-20 Ivo Swartjes (UT) *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*

2010-19 Henriette Cramer (UvA) *People's Responses to Autonomous and Adaptive Systems*

2010-18 Charlotte Gerritsen (VU) *Caught in the Act: Investigating Crime by Agent-Based Simulation*

2010-17 Spyros Kotoulas (VU) *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*

2010-16 Sicco Verwer (TUD) *Efficient Identification of Timed Automata, theory and practice*

2010-15 Lianne Bodenstaff (UT) *Managing Dependency Relations in Inter-Organizational Models*

2010-14 Sander van Splunter (VU) *Automated Web Service Reconfiguration*

2010-13 Gianluigi Folino (RUN) *High Performance Data Mining using Bio-inspired techniques*

2010-12 Susan van den Braak (UU) *Sensemaking software for crime analysis*

2010-11 Adriaan Ter Mors (TUD) *The world according to MARP: Multi-Agent Route Planning*

2010-10 Rebecca Ong (UL) *Mobile Communication and Protection of Children*

2010-09 Hugo Kielman (UL) *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*

2010-08 Krzysztof Siewicz (UL) *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*

2010-07 Wim Fikkert (UT) *A Gesture interaction at a Distance*

2010-06 Sander Bakkes (UvT) *Rapid Adaptation of Video Game AI*

2010-05 Claudia Hauff (UT), *Predicting the Effectiveness of Queries and Retrieval Systems*

2010-04 Olga Kulyk (UT), *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*

2010-03 Joost Geurts (CWI), *A Document Engineering Model and Processing Framework for Multimedia documents*

2010-02 Ingo Wassink (UT), *Work flows in Life Science*

2010-01 Matthijs van Leeuwen (UU), *Patterns that Matter*

2009-46 Loredana Afanasiev (UvA), *Querying XML: Benchmarks and Recursion*

2009-45 Jilles Vreeken (UU), *Making Pattern Mining Useful*

2009-44 Roberto Santana Tapia (UT), *Assessing Business-IT Alignment in Networked Organizations*

2009-43 Virginia Nunes Leal Franqueira (UT), *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*

2009-42 Toine Bogers (UvT), *Recommender Systems for Social Bookmarking*

2009-41 Igor Berezhnyy (UvT), *Digital Analysis of Paintings*

2009-40 Stephan Raaijmakers (UvT), *Multinomial Language Learning: Investigations into the Geometry of Language*

2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin), *Service Substitution – A Behavioral Approach Based on Petri Nets*

2009-38 Riina Vuorikari (OU), *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*

2009-37 Hendrik Drachsler (OUN), *Navigation Support for Learners in Informal Learning Networks*

2009-36 Marco Kalz (OUN), *Placement Support for Learners in Learning Networks*

2009-35 Wouter Koelewijn (UL), *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*

2009-34 Inge van de Weerd (UU), *Advancing in Software Product Management: An Incremental Method Engineering Approach*

2009-33 Khiet Truong (UT), *How Does Real Affect Affect Affect Recognition In Speech?*

2009-32 Rik Farenhorst (VU) and Remco de Boer (VU), *Architectural Knowledge Management: Supporting Architects and Auditors*

2009-31 Sofiya Katrenko (UVA), *A Closer Look at Learning Relations from Text*

2009-30 Marcin Zukowski (CWI), *Balancing vectorized query execution with bandwidth-optimized storage*

2009-29 Stanislav Pokraev (UT), *Model-Driven Semantic Integration of Service-Oriented Applications*

2009-27 Christian Glahn (OU), *Contextual Support of social Engagement and Reflection on the Web*

2009-26 Fernando Koch (UU), *An Agent-Based Model for the Development of Intelligent Mobile Services*

2009-25 Alex van Ballegooij (CWI), *RAM: Array Database Management through Relational Mapping*

2009-24 Annerieke Heuvelink (VUA), *Cognitive Models for Training Simulations*

2009-23 Peter Hofgesang (VU), *Modelling Web Usage in a Changing Environment*

2009-22 Pavel Serdyukov (UT), *Search For Expertise: Going beyond direct evidence*

2009-21 Stijn Vanderlooy (UM), *Ranking and Reliable Classification*

2009-20 Bob van der Vecht (UU), *Adjustable Autonomy: Controling Influences on Decision Making*

2009-19 Valentin Robu (CWI), *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*

2009-18 Fabian Groffen (CWI), *Armada, An Evolving Database System*

2009-17 Laurens van der Maaten (UvT), *Feature Extraction from Visual Data*

2009-16 Fritz Reul (UvT), *New Architectures in Computer Chess*

2009-15 Rinke Hoekstra (UVA), *Ontology Representation - Design Patterns and Ontologies that Make Sense*

2009-14 Maksym Korotkiy (VU), *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*

2009-13 Steven de Jong (UM), *Fairness in Multi-Agent Systems*

2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin), *Operating Guidelines for Services*

2009-11 Alexander Boer (UVA), *Legal Theory, Sources of Law & the Semantic Web*

2009-10 Jan Wielemaker (UVA), *Logic programming for knowledge-intensive interactive applications*

2009-09 Benjamin Kanagwa (RUN), *Design, Discovery and Construction of Service-oriented Systems*

2009-08 Volker Nannen (VU), *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*

2009-07 Ronald Poppe (UT), *Discriminative Vision-Based Recovery and Recognition of Human Motion*

2009-06 Muhammad Subianto (UU), *Understanding Classification*

2009-05 Sietse Overbeek (RUN), *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*

2009-04 Josephine Nabukenya (RUN), *Improving the Quality of Organisational Policy Making using Collaboration Engineering*

2009-03 Hans Stol (UvT), *A Framework for Evidence-based Policy Making Using IT*

2009-02 Willem Robert van Hage (VU), *Evaluating Ontology-Alignment Techniques*

2009-01 Rasa Jurgelenaite (RUN), *Symmetric Causal Independence Models*

2008-35 Ben Torben Nielsen (UvT), *Dendritic morphologies: function shapes structure*

2008-34 Jeroen de Knijf (UU), *Studies in Frequent Tree Mining*

2008-33 Frank Terpstra (UVA), *Scientific Workflow Design; theoretical and practical issues*

2008-32 Trung H. Bui (UT), *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*

2008-31 Loes Braun (UM), *Pro-Active Medical Information Retrieval*

2008-30 Wouter van Atteveldt (VU), *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*

2008-29 Dennis Reidsma (UT), *Annotations and Subjective Machines – Of Annotators, Embodied Agents, Users, and Other Humans*

2008-28 Ildiko Flesch (RUN), *On the Use of Independence Relations in Bayesian Networks*

2008-27 Hubert Vogten (OU), *Design and Implementation Strategies for IMS Learning Design*

2008-26 Marijn Huijbregts (UT), *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*

2008-25 Geert Jonker (UU), *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*

2008-24 Zharko Aleksovski (VU), *Using background knowledge in ontology matching*

2008-23 Stefan Visscher (UU), *Bayesian network models for the management of ventilator-associated pneumonia*

2008-22 Henk Koning (UU), *Communication of IT-Architecture*

2008-21 Krisztian Balog (UVA), *People Search in the Enterprise*

2008-20 Rex Arendsen (UVA), *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven.*

2008-19 Henning Rode (UT), *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*

2008-18 Guido de Croon (UM), *Adaptive Active Vision*

2008-17 Martin Op 't Land (TUD), *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*

2008-16 Henriëtte van Vugt (VU), *Embodied agents from a user's perspective*

2008-15 Martijn van Otterlo (UT), *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.*

2008-14 Arthur van Bunningen (UT), *Context-Aware Querying; Better Answers with Less Effort*

2008-13 Caterina Carraciolo (UVA), *Topic Driven Access to Scientific Handbooks*

2008-12 József Farkas (RUN), *A Semiotically Oriented Cognitive Model of Knowledge Representation*

2008-11 Vera Kartseva (VU), *Designing Controls for Network Organizations: A Value-Based Approach*

2008-10 Wauter Bosma (UT), *Discourse oriented summarization*

2008-09 Christof van Nimwegen (UU), *The paradox of the guided user: assistance can be counter-effective*

2008-08 Janneke Bolt (UU), *Bayesian Networks: Aspects of Approximate Inference*

2008-07 Peter van Rosmalen (OU), *Supporting the tutor in the design and support of adaptive e-learning*

2008-06 Arjen Hommersom (RUN), *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*

2008-05 Bela Mutschler (UT), *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*

2008-04 Ander de Keijzer (UT), *Management of Uncertain Data – towards unattended integration*

2008-03 Vera Hollink (UVA), *Optimizing hierarchical menus: a usage-based approach*

2008-02 Alexei Sharpanskykh (VU), *On Computer-Aided Methods for Modeling and Analysis of Organizations*

2008-01 Katalin Boer-Sorbán (EUR), *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*

2007-25 Joost Schalken (VU), *Empirical Investigations in Software Process Improvement*

2007-24 Georgina Ramírez Camps (CWI), *Structural Features in XML Retrieval*

2007-23 Peter Barna (TUE), *Specification of Application Logic in Web Information Systems*

2007-22 Zlatko Zlatev (UT), *Goal-oriented design of value and process models from patterns*

2007-21 Karianne Vermaas (UU), *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*

2007-20 Slinger Jansen (UU), *Customer Configuration Updating in a Software Supply Network*

2007-19 David Levy (UM), *Intimate relationships with artificial partners*

2007-18 Bart Orriëns (UvT), *On the development an management of adaptive business collaborations*

2007-17 Theodore Charitos (UU), *Reasoning with Dynamic Networks in Practice*

2007-16 Davide Grossi (UU), *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*

2007-15 Joyca Lacroix (UM), *NIM: a Situated Computational Memory Model*

2007-14 Niek Bergboer (UM), *Context-Based Image Analysis*

2007-13 Rutger Rienks (UT), *Meetings in Smart Environments; Implications of Progressing Technology*

2007-12 Marcel van Gerven (RUN), *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*

2007-11 Natalia Stash (TUE), *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*

2007-10 Huib Aldewereld (UU), *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*

2007-09 David Mobach (VU), *Agent-Based Mediated Service Negotiation*

2007-08 Mark Hoogendoorn (VU), *Modeling of Change in Multi-Agent Organizations*

2007-07 Natasa Jovanović (UT), *To Whom It May Concern – Addressee Identification in Face-to-Face Meetings*

2007-06 Gilad Mishne (UVA), *Applied Text Analytics for Blogs*

2007-05 Bart Schermer (UL), *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*

2007-04 Jurriaan van Diggelen (UU), *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*

2007-03 Peter Mika (VU), *Social Networks and the Semantic Web*

2007-02 Wouter Teepe (RUG), *Reconciling Information Exchange and Confidentiality: A Formal Approach*

2007-01 Kees Leune (UvT), *Access Control and Service-Oriented Architectures*

2006-28 Börkur Sigurbjörnsson (UVA), *Focused Information Access using XML Element Retrieval*

2006-27 Stefano Bocconi (CWI), *Vox Populi: generating video documentaries from semantically annotated media repositories*

2006-26 Vojkan Mihajlović (UT), *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
2006-25 Madalina Drugan (UU), *Conditional log-likelihood MDL and Evolutionary MCMC*
2006-24 Laura Hollink (VU), *Semantic Annotation for Retrieval of Visual Resources*
2006-23 Ion Juvina (UU), *Development of Cognitive Model for Navigating on the Web*
2006-22 Paul de Vrieze (RUN), *Fundaments of Adaptive Personalisation*
2006-21 Bas van Gils (RUN), *Aptness on the Web*
2006-20 Marina Velikova (UvT), *Monotone models for prediction in data mining*
2006-19 Birna van Riemsdijk (UU), *Cognitive Agent Programming: A Semantic Approach*
2006-18 Valentin Zhizhkun (UVA), *Graph transformation for Natural Language Processing*
2006-17 Stacey Nagata (UU), *User Assistance for Multitasking with Interruptions on a Mobile Device*
2006-16 Carsten Riggelsen (UU), *Approximation Methods for Efficient Learning of Bayesian Networks*
2006-15 Rainer Malik (UU), *CONAN: Text Mining in the Biomedical Domain*
2006-14 Johan Hoorn (VU), *Software Requirements: Update, Upgrade, Redesign – towards a Theory of Requirements Change*
2006-13 Henk-Jan Lebbink (UU), *Dialogue and Decision Games for Information Exchanging Agents*
2006-12 Bert Bongers (VU), *Interactivation – Towards an e-cology of people, our technological environment, and the arts*
2006-11 Joeri van Ruth (UT), *Flattening Queries over Nested Data Types*
2006-10 Ronny Siebes (VU), *Semantic Routing in Peer-to-Peer Systems*
2006-09 Mohamed Wahdan (UM), *Automatic Formulation of the Auditor's Opinion*
2006-08 Eelco Herder (UT), *Forward, Back and Home Again – Analyzing User Behavior on the Web*
2006-07 Marko Smiljanic (UT), *XML schema matching – balancing efficiency and effectiveness by means of clustering*
2006-06 Ziv Baida (VU), *Software-aided Service Bundling – Intelligent Methods & Tools for Graphical Service Modeling*
2006-05 Cees Pierik (UU), *Validation Techniques for Object-Oriented Proof Outlines*
2006-04 Marta Sabou (VU), *Building Web Service Ontologies*
2006-03 Noor Christoph (UVA), *The role of metacognitive skills in learning to solve problems*
2006-02 Cristina Chisalita (VU), *Contextual issues in the design and use of information technology in organizations*
2006-01 Samuil Angelov (TUE), *Foundations of B2B Electronic Contracting*
2005-21 Wijnand Derks (UT), *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*
2005-20 Cristina Coteanu (UL), *Cyber Consumer Law, State of the Art and Perspectives*
2005-19 Michel van Dartel (UM), *Situated Representation*
2005-18 Danielle Sent (UU), *Test-selection strategies for probabilistic networks*
2005-17 Boris Shishkov (TUD), *Software Specification Based on Re-usable Business Components*
2005-16 Joris Graaumans (UU), *Usability of XML Query Languages*
2005-15 Tibor Bosse (VU), *Analysis of the Dynamics of Cognitive Processes*
2005-14 Borys Omelayenko (VU), *Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics*
2005-13 Fred Hamburg (UL), *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*
2005-12 Csaba Boer (EUR), *Distributed Simulation in Industry*
2005-11 Elth Ogston (VU), *Agent Based Matchmaking and Clustering – A Decentralized Approach to Search*
2005-10 Anders Bouwer (UVA), *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*
2005-09 Jeen Broekstra (VU), *Storage, Querying and Inferencing for Semantic Web Languages*
2005-08 Richard Vdovjak (TUE), *A Model-driven Approach for Building Distributed Ontology-based Web Applications*
2005-07 Flavius Frasincar (TUE), *Hypermedia Presentation Generation for Semantic Web Information Systems*
2005-06 Pieter Spronck (UM), *Adaptive Game AI*
2005-05 Gabriel Infante-Lopez (UVA), *Two-Level Probabilistic Grammars for Natural Language Parsing*
2005-04 Nirvana Meratnia (UT), *Towards Database Support for Moving Object data*
2005-03 Franc Grootjen (RUN), *A Pragmatic Approach to the Conceptualisation of Language*
2005-02 Erik van der Werf (UM)), *AI techniques for the game of Go*
2005-01 Floor Verdenius (UVA), *Methodological Aspects of Designing Induction-Based Applications*
2004-20 Madelon Evers (Nyenrode), *Learning from Design: facilitating multidisciplinary design teams*
2004-19 Thijs Westerveld (UT), *Using generative probabilistic models for multimedia retrieval*
2004-18 Vania Bessa Machado (UvA), *Supporting the Construction of Qualitative Knowledge Models*
2004-17 Mark Winands (UM), *Informed Search in Complex Games*
2004-16 Federico Divina (VU), *Hybrid Genetic Relational Search for Inductive Learning*
2004-15 Arno Knobbe (UU), *Multi-Relational Data Mining*
2004-14 Paul Harrenstein (UU), *Logic in Conflict. Logical Explorations in Strategic Equilibrium*
2004-13 Wojciech Jamroga (UT), *Using Multiple Models of Reality: On Agents who Know how to Play*
2004-12 The Duy Bui (UT), *Creating emotions and facial expressions for embodied agents*
2004-11 Michel Klein (VU), *Change Management for Distributed Ontologies*
2004-10 Suzanne Kabel (UVA), *Knowledge-rich indexing of learning-objects*

2004-09 Martin Caminada (VU), *For the Sake of the Argument; explorations into argument-based reasoning*

2004-08 Joop Verbeek (UM), *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise*

2004-07 Elise Boltjes (UM), *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*

2004-06 Bart-Jan Hommes (TUD), *The Evaluation of Business Process Modeling Techniques*

2004-05 Viara Popova (EUR), *Knowledge discovery and monotonicity*

2004-04 Chris van Aart (UVA), *Organizational Principles for Multi-Agent Architectures*

2004-03 Perry Groot (VU), *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*

2004-02 Lai Xu (UvT), *Monitoring Multi-party Contracts for E-business*

2004-01 Virginia Dignum (UU), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*

2003-18 Levente Kocsis (UM), *Learning Search Decisions*

2003-17 David Jansen (UT), *Extensions of Statecharts with Probability, Time, and Stochastic Timing*

2003-16 Menzo Windhouwer (CWI), *Feature Grammar Systems – Incremental Maintenance of Indexes to Digital Media Warehouses*

2003-15 Mathijs de Weerdt (TUD), *Plan Merging in Multi-Agent Systems*

2003-14 Stijn Hoppenbrouwers (KUN), *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*

2003-13 Jeroen Donkers (UM), *Nosce Hostem – Searching with Opponent Models*

2003-12 Roeland Ordelman (UT), *Dutch speech recognition in multimedia information retrieval*

2003-11 Simon Keizer (UT), *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*

2003-10 Andreas Lincke (UvT), *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*

2003-09 Rens Kortmann (UM), *The resolution of visually guided behaviour*

2003-08 Yongping Ran (UM), *Repair Based Scheduling*

2003-07 Machiel Jansen (UvA), *Formal Explorations of Knowledge Intensive Tasks*

2003-06 Boris van Schooten (UT), *Development and specification of virtual environments*

2003-05 Jos Lehmann (UVA), *Causation in Artificial Intelligence and Law – A modelling approach*

2003-04 Milan Petković (UT), *Content-Based Video Retrieval Supported by Database Technology*

2003-03 Martijn Schuemie (TUD), *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*

2003-02 Jan Broersen (VU), *Modal Action Logics for Reasoning About Reactive Systems*

2003-01 Heiner Stuckenschmidt (VU), *Ontology-Based Information Sharing in Weakly Structured Environments*

2002-17 Stefan Manegold (UVA), *Understanding, Modeling, and Improving Main-Memory Database Performance*

2002-16 Pieter van Langen (VU), *The Anatomy of Design: Foundations, Models and Applications*

2002-15 Rik Eshuis (UT), *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*

2002-14 Wieke de Vries (UU), *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

2002-13 Hongjing Wu (TUE), *A Reference Architecture for Adaptive Hypermedia Applications*

2002-12 Albrecht Schmidt (Uva), *Processing XML in Database Systems*

2002-11 Wouter C.A. Wijngaards (VU), *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

2002-10 Brian Sheppard (UM), *Towards Perfect Play of Scrabble*

2002-09 Willem-Jan van den Heuvel (KUB), *Integrating Modern Business Applications with Objectified Legacy Systems*

2002-08 Jaap Gordijn (VU), *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*

2002-07 Peter Boncz (CWI), *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*

2002-06 Laurens Mommers (UL), *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*

2002-05 Radu Serban (VU), *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*

2002-04 Juan Roberto Castelo Valdueza (UU), *The Discrete Acyclic Digraph Markov Model in Data Mining*

2002-03 Henk Ernst Blok (UT), *Database Optimization Aspects for Information Retrieval*

2002-02 Roelof van Zwol (UT), *Modelling and searching web-based document collections*

2002-01 Nico Lassing (VU), *Architecture-Level Modifiability Analysis*

2001-11 Tom M. van Engers (VUA), *Knowledge Management: The Role of Mental Models in Business Systems Design*

2001-10 Maarten Sierhuis (UvA), *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*

2001-09 Pieter Jan 't Hoen (RUL), *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*

2001-08 Pascal van Eck (VU), *A Compositional Semantic Structure for Multi-Agent Systems Dynamics.*

2001-07 Bastiaan Schonhage (VU), *Diva: Architectural Perspectives on Information Visualization*

2001-06 Martijn van Welie (VU), *Task-based User Interface Design*

2001-05 Jacco van Ossenbruggen (VU), *Processing Structured Hypermedia: A Matter of Style*

2001-04 Evgueni Smirnov (UM), *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*

2001-03 Maarten van Someren (UvA), *Learning as problem solving*

2001-02 Koen Hindriks (UU), *Agent Programming Languages: Programming with Mental Models*

2001-01 Silja Renooij (UU), *Qualitative Approaches to Quantifying Probabilistic Networks*

2000-11 Jonas Karlsson (CWI), *Scalable Distributed Data Structures for Database Management*

2000-10 Niels Nes (CWI), *Image Database Management System Design Considerations, Algorithms and Architecture*

2000-09 Florian Waas (CWI), *Principles of Probabilistic Query Optimization*

2000-08 Veerle Coupé (EUR), *Sensitivity Analyis of Decision-Theoretic Networks*

2000-07 Niels Peek (UU), *Decision-theoretic Planning of Clinical Patient Management*

2000-06 Rogier van Eijk (UU), *Programming Languages for Agent Communication*

2000-05 Ruud van der Pol (UM), *Knowledge-based Query Formulation in Information Retrieval.*

2000-04 Geert de Haan (VU), *ETAG, A Formal Model of Competence Knowledge for User Interface Design*

2000-03 Carolien M.T. Metselaar (UVA), *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.*

2000-02 Koen Holtman (TUE), *Prototyping of CMS Storage Management*

2000-01 Frank Niessink (VU), *Perspectives on Improving Software Maintenance*

1999-08 Jacques H.J. Lenting (UM), *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.*

1999-07 David Spelt (UT), *Verification support for object database design*

1999-06 Niek J.E. Wijngaards (VU), *Re-design of compositional systems*

1999-05 Aldo de Moor (KUB), *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*

1999-04 Jacques Penders (UM), *The practical Art of Moving Physical Objects*

1999-03 Don Beal (UM), *The Nature of Minimax Search*

1999-02 Rob Potharst (EUR), *Classification using decision trees and neural nets*

1999-01 Mark Sloof (VU), *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*

1998-05 E.W. Oskamp (RUL), *Computerondersteuning bij Straftoemeting*

1998-04 Dennis Breuker (UM), *Memory versus Search in Games*

1998-03 Ans Steuten (TUD), *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*

1998-02 Floris Wiesman (UM), *Information Retrieval by Graphically Browsing Meta-Information*

1998-01 Johan van den Akker (CWI), *DEGAS – An Active, Temporal Database of Autonomous Objects*

Bibliography

J. Adcock. Fxpal interacive search experiments for TRECVid 2007. In *Proceedings of the 7th TRECVid Workshop*, Gaithersburg, USA, October 2007.

S. M. Aji and R. J. McEliece. The generalized distributive law. *Information Theory, IEEE Transactions on*, 46(2):325–343, 2000. ISSN 0018-9448. doi: 10.1109/18.825794.

R. Aly. Modeling uncertainty in video retrieval. In *SIGIR '09: Proceedings of the 32th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 846–846, Boston, 2009. ACM. ISBN 978-1-60558-483-6. doi: http://doi.acm.org/10.1145/1571941.1572167.

R. Aly and D. Hiemstra. Concept detectors: how good is good enough? In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 233–242, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-608-3. doi: http://doi.acm.org/10.1145/1631272.1631306.

R. Aly and D. Hiemstra. A simulator for concept detector output. Technical Report TR-CTIT-09-40, University Twente, Enschede, December 2009b. URL `http://eprints.eemcs.utwente.nl/16544/`.

R. Aly, C. Hauff, W. Heeren, D. Hiemstra, F. de Jong, R. Ordelman, T. Verschoor, and A. de Vries. The lowlands team at TRECVid 2007. In *Proceedings of the 7th TRECVid Workshop*, Geithesburg, U.S., February 2007a. NIST. ISBN not assigned.

R. Aly, D. Hiemstra, and R. Ordelman. Building detectors to support searches on combined semantic concepts. In *Proceedings of the SIGIR Multimedia Information Retrieval Workshop, Amsterdam, The Netherlands*, pages 40–45, Amsterdam, August 2007b. Yahoo! Research.

R. Aly, D. Hiemstra, R. Ordelman, L. van der Werff, and F. de Jong. XML Information Retrieval from Spoken Word Archives. In *Evaluation of Multilingual and Multi-modal Information Retrieval*, volume 4730 of *Lecture Notes in Computer Science*, pages 770–777, Berlin, September 2007c. Springer Verlag.

R. Aly, D. Hiemstra, A. P. de Vries, and F. de Jong. A probabilistic ranking framework using unobservable binary events for video search. In *CIVR '08: Proceedings of the International Conference on Content-Based Image and Video Retrieval 2008*, pages 349–358, 2008a. ISBN 978-1-60558-070-8. doi: http://doi.acm.org/10.1145/1386352.1386398.

R. Aly, D. Hiemstra, A. P. de Vries, and H. Rode. The lowlands team at TRECVid 2008. In *Proceedings of the 8th TRECVid Workshop*, 2008b.

R. Aly, D. Hiemstra, and A. P. de Vries. Reusing annotation labor for concept selection. In *CIVR '09: Proceedings of the International Conference on Content-Based Image and Video Retrieval.* ACM, 2009. ISBN 978-1-60558-070-8.

R. Aly, A. Doherty, D. Hiemstra, and A. Smeaton. Beyond shot retrieval: Searching for broadcast news items using language models of concepts. In *ECIR '10: Proceedings of the 32th European Conference on IR Research on Advances in Information Retrieval*, volume 5993 of *Lecture Notes in Computer Science*, pages 241–252, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 978-3-642-00957-0.

M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 971–980, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: http://doi.acm.org/10.1145/1240624.1240772.

C. Arndt. *Information Measures: Information and its description in Science and Engineering.* Springer, 2001.

J. A. Aslam and M. Montague. Models for metasearch. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: http://doi.acm.org/10.1145/383952.384007.

S. Ayache and G. Quénot. Video corpus annotation using active learning. In *30h European Conference on Information Retrieval (ECIR'08)*, pages 187–198, March 30 2008.

R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999. ISBN 0-201-39829-X.

J. Bather. *Decision Theory. An Introduction to Dynamic Programming and Sequential Decisions.* Wiley-Interscience Series in Systems and Optimisation. John Wiley and Sons, West Sussex, England, 2000.

E. Berry, A. Hampshire, J. Rowe, S. Hodges, N. Kapur, P. Watson, G. Browne, G. Smyth, K. Wood, and A. M. Owen. The neural basis of effective memory therapy in a patient with limbic encephalitis. *Journal of Neurology, Neurosurgery & Psychiatry*, 80(11):1202–1205, November 2009. doi: 10.1136/jnnp.2008.164251. URL `http://dx.doi.org/10.1136/jnnp.2008.164251`.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006. ISBN 0387310738.

D. Bodoff and S. E. Robertson. A new unified probabilistic model. *J. Am. Soc. Inf. Sci. Technol.*, 55(6):471–487, April 2004. ISSN 1532-2882. doi: http://dx.doi.org/10.1002/asi.10398. URL `http://dx.doi.org/10.1002/asi.10398`.

A. C. Bovik, M. Clark, W. S. Geisler, A. Bovik, M. Clark, and W. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1)(1):55–73, Jan 1990. ISSN 0162-8828. doi: 10.1109/34.41384.

S.-F. Chang, W. Hsu, W. Jiang, L. Kennedy, D. Xu, A. Yanagawa, and E. Zavesky. Columbia university trecvid-2006 video search and high-level feature extraction. In *Proceedings of the 4th TRECVid Workshop*, Gaithersburg, USA, November 2006.

C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 443–450, Morristown, NJ, USA, 2005. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1219840.1219895.

T. K. Chia, K. C. Sim, H. Li, and H. T. Ng. A lattice-based approach to query-by-example spoken document retrieval. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 363–370, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: http://doi.acm.org/10.1145/1390334.1390397.

G. Chowdhury. *Introduction to modern information retrieval.* John Wiley & Sons, 1998.

M. G. Christel and A. G. Hauptmann. The use and utility of high-level semantic features in video retrieval. In *Image and Video Retrieval*, volume Volume 3568/2005, pages 134–144. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-27858-0. doi: 10.1007/1152634617. URL `http://www.springerlink.com/content/1mf31v38vgltkg1r/`.

W. B. Croft. Document representations in probabilistic models of information retrieval. *Journal of the American Society of Information Science*, 32(6): 451–457, 1981.

W. B. Croft. Experiments with representation in a document retrieval system. *Information Technology*, 2(1):1–21, 1983.

W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35(4): 285–295, 1979.

W. B. Croft, S. Harding, K. Taghva, and J. Borsack. An evaluation of information retrieval accuracy with simulated ocr output. In *In Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, pages 115–126, 1992.

K. M. Donald and A. F. Smeaton. A comparison of score, rank and probability-based fusion methods for video shot retrieval. In *Image and Video Retrieval*, volume Volume 3568/2005, pages 61–70. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-27858-0. doi: 10.1007/1152634610. URL `http://www.springerlink.com/content/9jwatefm7p00dmkm/`.

K.-B. Duan and S. S. Keerthi. Which is the best multiclass SVM method? an empirical study. In *Multiple Classifier Systems*, volume 3541 of *Lecture Notes in Computer Science*, pages 278–285. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-26306-7. doi: 10.1007/11494683_28. URL `http://www.springerlink.com/content/r9deb9rv9x5qdxjc/`.

T. Dunning. Accurate methods for the statistics of surprise and coincidence. *T. Dunning.*, 19(1):61–74, 1993.

P. Enser. Progress in documentation pictorial information retrieval. *Journal of Documentation*, 51(2):126–170, 1995. doi: 10.1108/eb026946.

C. Fellbaum. *Wordnet: An Electronic Lexical Database.* The MIT Press, 1998.

E. A. Fox and J. A. Shaw. Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2)*, pages 243–252, 1993.

N. Fuhr. Models for retrieval with probabilistic indexing. *Information Processing and Management*, 25(1):55–72, 1989.

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

A. Hanjalic. Shot-boundary detection: unraveled and resolved? *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(2):90–105, Feb 2002. ISSN 1051-8215. doi: 10.1109/76.988656.

D. Harman. Overview of the seventh text retrieval conference (trec-3). In D. Harman, editor, *Proceedings of the Seventh Text REtrieval Conference (TREC-3)*, pages 1–19, April 1995.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, page 147—151., 1988.

T. Hastie and R. Tibshirani. Classification by pairwise coupling. Technical report, Standford University and University of Torronto, 1996.

A. Haubold, A. Natsev, and M. R. Naphade. Semantic multimedia retrieval using lexical query expansion and model-based reranking. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1761–1764, 2006. URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4036961`.

C. Hauff, R. Aly, and D. Hiemstra. The effectiveness of concept based search for video retrieval. In *Workshop Information Retrieval (FGIR 2007), Halle, Germany*, volume 2007 of *LWA 2007: Lernen - Wissen - Adaption*, pages 205–212, Halle-Wittenberg, 2007. Gesellschaft fuer Informatik.

A. G. Hauptmann, R. Yan, W.-H. Lin, M. Christel, and H. Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. In *IEEE Transactions on Multimedia*, volume 9-5, pages 958–966, August 2007. doi: 10.1109/TMM.2007.900150.

D. Hawking. Overview of the trec-9 web track. In E. M. Voorhees and D. K. Harman, editors, *NIST Special Publication 500-249: The Ninth Text REtrieval Conference (TREC 9)*, page 87, 2000.

W. F. L. Heeren, L. B. van der Werff, F. M. G. de Jong, R. J. F. Ordelman, T. Verschoor, A. J. van Hessen, and M. Langelaar. Easy listening: Spoken document retrieval in choral. *Interdisciplinary Science Reviews*, 34(2-3): 236–252, September 2009. ISSN 0308-0188.

D. Hiemstra. *Using Language Models for Information Retrieval.* PhD thesis, University of Twente, Enschede, January 2001.

D. Hiemstra, H. Rode, T. van Os, Roel, and J. Flokstra. Pftijah: text search in an xml database system. In *Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR), Seattle, WA, USA*, pages 12–17. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2006.

W. H. Hsu, L. S. Kennedy, and S.-F. Chang. Video search reranking via information bottleneck principle. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 35–44, New York, NY, USA, 2006. ACM. ISBN 1-59593-447-2. doi: http://doi.acm.org/10.1145/1180639.1180654.

M. A. H. Huijbregts. *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled.* PhD thesis, Univ. of Twente, Enschede, November 2008.

B. Huurnink. AutoSeek: Towards a fully automated video search system. Master's thesis, University of Amsterdam, October 2005.

B. Huurnink, K. Hofmann, and M. de Rijke. Assessing concept selection for video retrieval. In *Proceedings of the first MIR Conference'08*, Oktober 2008.

Y.-G. Jiang, C.-W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 494–501, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-733-9. doi: http://doi.acm.org/10.1145/1282280.1282352.

Y.-G. Jiang, J. Yang, C.-W. Ngo, and A. Hauptmann. Representations of keypoint-based semantic concept detection: A comprehensive study. *Multimedia, IEEE Transactions on*, 12(1):42 –53, jan. 2010. ISSN 1520-9210. doi: 10.1109/TMM.2009.2036235.

L. Kennedy, S.-F. Chang, and A. Natsev. Query-adaptive fusion for multimodal search. *Proceedings of the IEEE*, 96(4):567–588, April 2008. ISSN 0018-9219. doi: 10.1109/JPROC.2008.916345.

L. S. Kennedy, A. P. Natsev, and S.-F. Chang. Automatic discovery of query-class-dependent models for multimodal search. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 882–891, New York, NY, USA, 2005. ACM. ISBN 1-59593-044-2. doi: http://doi.acm.org/10.1145/1101149.1101339.

V. Lavrenko. *A generative theory of relevance.* PhD thesis, University of Edinburgh, 2004. URL `http://portal.acm.org/citation.cfm?id=1087151`.

X. Li, D. Wang, J. Li, and B. Zhang. Video search in concept subspace: a text-like paradigm. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 603–610, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-733-9. doi: http://doi.acm.org/10.1145/1282280.1282366.

H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3): 267–276, October 2007. ISSN 0885-6125 (Print) 1573-0565 (Online). doi: 10.1007/s10994-007-5018-6. URL `http://www.springerlink.com/content/8417v9235m561471/`.

J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer, October 2002. ISBN 0387952306.

S. Loh, L. K. Wives, and J. P. M. de Oliveira. Concept-based knowledge discovery in texts extracted from the web. *SIGKDD Explor. Newsl.*, 2(1): 29–39, 2000. ISSN 1931-0145. doi: http://doi.acm.org/10.1145/360402.360414.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004. ISSN 0920-5691 (Print) 1573-1405 (Online). doi: 10.1023/B:VISI.0000029664.99615.94. URL `http://www.springerlink.com/content/h4l02691327px768/`.

G. Lu. Indexing and retrieval of audio: A survey. *Multimedia Tools and Applications*, 15(3):269–290, December 2001. ISSN 1380-7501 (Print) 1573-7721 (Online). doi: 10.1023/A:1012491016871. URL `http://www.springerlink.com/content/r0032342x664764p/`.

J. Mamou, D. Carmel, and R. Hoory. Spoken document retrieval from call-center conversations. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 51–58, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: http://doi.acm.org/10.1145/1148170.1148183.

M. Markkula and E. Sormunen. End-user searching challenges indexing practices in the digital newspaper photo archive. *Information Retrieval*, 1 (4):259–285, January 2000. ISSN 1386-4564 (Print) 1573-7659 (Online). doi: 10.1023/A:1009995816485. URL `http://www.springerlink.com/content/j3440h3nu4422235/`.

H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. ISSN 00221082. URL `http://www.jstor.org/stable/2975974`.

M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *J. ACM*, 7(3):216–244, 1960. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/321033.321035.

N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949. ISSN 01621459. URL `http://www.jstor.org/stable/2280232`.

R. G. Millikan. *On Clear and Confused Ideas: An Essay about Substance Concepts.* Cambridge University Press, July 2000. ISBN 052162553X.

R. Moreno and R. E. Mayer. Cognitive principles of multimedia learning: The role of modality and contiguity. *Journal of Educational Psychology*, 91(2):358–368, 1999.

M. Naphade, J. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. G. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86–91, 2006. ISSN 1070-986X. doi: 10.1109/MMUL.2006.63.

M. R. Naphade and J. R. Smith. On the detection of semantic concepts at trecvid. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 660–667, New York, NY, USA, 2004. ACM. ISBN 1-58113-893-8. doi: http://doi.acm.org/10.1145/1027527.1027680.

A. P. Natsev, A. Haubold, J. Tevsi'c, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 991–1000, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-702-5. doi: http://doi.acm.org/10.1145/1291233.1291448.

R. Ordelman, F. de Jong, and D. van Leeuwen. *Multimedia Retrieval. Data-Centric Systems and Applications*, chapter Speech Indexing, pages 199–224. Springer Verlag, 2007.

A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw Hill, 1984.

Y. Peng, Z. Lu, and J. Xiao. Semantic concept annotation based on audio plsa model. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 841–844, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-608-3. doi: http://doi.acm.org/10.1145/1631272.1631428.

C. Petersohn. Fraunhofer hhi at TRECVid 2004: Shot boundary detection system. In *Proceedings of the 3rd TRECVid Workshop*, 2004. URL `www-nlpir.nist.gov/projects/tvpubs/tvpapers04/fraunhofer.pdf`.

J. Platt. *Advances in Large Margin Classifiers*, chapter Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, pages 61–74. MIT Press, Cambridge, MA, 2000.

J. M. Ponte. *A language modeling approach to information retrieval.* PhD thesis, University of Massachusetts Amherst, 1998.

J. Portelo, M. Bugalho, I. Trancoso, J. Neto, A. Abad, and A. Serralheiro. Non-speech audio event detection. In *ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1973–1976, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-1-4244-2353-8. doi: http://dx.doi.org/10.1109/ICASSP.2009.4959998.

W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C, The Art of Scientific Computing.* Cambridge University Press use, 2nd edition, 1992.

S. E. Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33:294–304, 1977.

S. E. Robertson. On event spaces and probabilistic models in information retrieval. *Information Retrieval*, 8(2):319–329, April 2005. ISSN 1386-4564 (Print) 1573-7659 (Online). doi: 10.1007/s10791-005-5665-9. URL `http://www.springerlink.com/content/t7l48g5u6w424033`.

S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X.

S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 35–56, Kent, UK, 1981. Butterworth & Co. ISBN 0-408-10775-8.

S. E. Robertson, M. E. Maron, and W. S. Cooper. Probability of relevance: A unification of two competing models for document retrieval. *Information Technology: Research and Development*, 1(1):1–21, January 1982.

K. Rodden, W. Basalaj, D. Sinclair, and K. Wood. Does organisation by similarity assist image browsing? In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 190–197, New York, NY, USA, 2001. ACM. ISBN 1-58113-327-8. doi: http://doi.acm.org/10.1145/365024.365097.

H. Rode and D. Hiemstra. Using query profiles for clarification. In M. Lalmas, A. MacFarlane, S. M. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, editors, *ECIR*, volume 3936 of *Lecture Notes in Computer Science*, pages 205–216. Springer, 2006. ISBN 3-540-33347-9.

S. M. Ross. *Introduction to Probability Models.* Academic Press, December 2006. ISBN 0125980620.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975. ISSN 0001-0782. doi: 10.1145/361219.361220. URL `http://dx.doi.org/10.1145/361219.361220`.

A. Sangswang and C. Nwankpa. Justification of a stochastic model for a dc-dc boost converter. In *Industrial Electronics Society, 2003. IECON '03. The 29th Annual Conference of the IEEE*, volume 2, pages 1870–1875 Vol.2, Nov. 2003. doi: 10.1109/IECON.2003.1280345.

J. Sivic and A. 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*, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1950-4. URL `http://portal.acm.org/citation.cfm?id=946247.946751`.

A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-495-2. doi: http://doi.acm.org/10.1145/1178677.1178722.

A. F. Smeaton, P. Over, and A. R. Doherty. Video shot boundary detection: Seven years of TRECVid activity. *Computer Vision and Image Understanding*, In Press, Corrected Proof:–, 2009. ISSN 1077-3142. doi: DOI:10.1016/j.cviu.2009.03.011. URL `http://www.sciencedirect.com/science/article/B6WCX-4VXMPVJ-1/2/5ee5382b7330215937a8dd430669dc8f`.

A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.895972.

C. G. M. Snoek and M. Worring. Are concept detector lexicons effective for video search? In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1966–1969, 2007. doi: 10.1109/ICME.2007.4285063.

C. G. M. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 4(2):215–322, 2009.

C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-447-2. doi: http://doi.acm.org/10.1145/1180639.1180727.

C. G. M. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring. Adding semantics to detectors for video retrieval. *IEEE Transactions on Multimedia*, 9(5):975–986, 2007.

C. G. M. Snoek, K. van de Sande, O. de Rooij, B. Huurnink, J. van Gemert, J. Uijlings, J. He, X. Li, I. Everts, V. Nedovic, M. van Liempt, R. van Balen, M. de Rijke, J. Geusebroek, T. Gevers, M. Worring, A. Smeulders, D. Koelma, F. Yan, M. Tahir, K. Mikolajczyk, and J. Kittler. The Media-Mill TRECVid 2008 semantic video search engine. In *Proceedings of the 8th TRECVid Workshop*, Gaithersburg, USA, November 2008.

K. Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–20, 1972. URL http://portal.acm.org/citation.cfm?id=106782.

K. Spärck-Jones and P. Willett, editors. *Readings in information retrieval.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1-55860-454-5.

K. Spärck-Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 2. *Information Processing and Management*, 36(6):809–840, 2000.

B. Stroustrup. *The C++ Programming Language.* Addison-Wesley Professional, third edition, February 2000. ISBN 0201700735. URL http://www.worldcat.org/isbn/0201700735.

J. R. Taylor. *An Introduction to Error Analysis.* University Science Books, 2 sub edition, 1996. ISBN 093570275X.

R. S. Taylor. The process of asking questions. *American Documentation*, 13 (4):391–396, 1962. ISSN 1936-6108. doi: 10.1002/asi.5090130405. URL http://dx.doi.org/10.1002/asi.5090130405.

P. Toharia, O. D. Robles, A. F. Smeaton, and A. Rodríguez. Measuring the influence of concept detection on video retrieval. In *CAIP 2009 - 13th International Conference on Computer Analysis of Images and Patterns.* Springer, September 2009.

D. Trieschnigg, P. Pezik, V. Lee, W. Kraaij, F. de Jong, and D. Rebholz-Schuhmann. MeSH Up: Effective MeSH Text Classification and Improved Document Retrieval. *Bioinformatics*, 25(11):1412–1418, 2009.

M. F. Triola. *Essentials of Statistics.* Addison Wesley, 2008.

K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, X(in press): X, 2010. URL http://www.science.uva.nl/research/publications/2010/vandeSandeTPAMI2010.

V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, NY, November 1999. ISBN 0387987800.

E. M. Voorhees and D. Harman. Overview of the ninth text retrieval conference (trec-9). In *In Proceedings of the Ninth Text REtrieval Conference (TREC-9*, pages 1–14, 2000.

A. P. d. Vries, G. Kazai, and M. Lalmas. Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit. In *RIAO 2004 Conference Proceedings*, pages 463–473, Avignon, France, April 2004.

J. Wang. Mean-variance analysis: A new document ranking theory in information retrieval. In *ECIR '09: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 4–16, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-00957-0. doi: http://dx.doi.org/10.1007/978-3-642-00958-7_4. URL `http://www.springerlink.com/content/635330kl602h1775/`.

J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: http://doi.acm.org/10.1145/1571941.1571963.

M. Wang, X. Zhou, and T.-S. Chua. Automatic image annotation via local multi-label classification. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 17–26, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-070-8. doi: http://doi.acm.org/10.1145/1386352.1386359.

X. Y. Wei, C. W. Ngo, and Y. G. Jiang. Selection of concept detectors for video search by ontology-enriched semantic spaces. *IEEE Trans. on Multimedia*, 10(6):1085–1096, 2008.

X.-Y. Wei, Y.-G. Jiang, and C.-W. Ngo. Exploring inter-concept relationship with context space for semantic video indexing. In *CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-480-5. doi: http://doi.acm.org/10.1145/1646396.1646416.

M. Witbrock and A. G. Hauptmann. Speech recognition and information retrieval: Experiments in retrieving spoken documents. In *In proceedings of the the DARAP Speech Recognition Workshop 1997*, pages 2–5, 1997.

Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, Morristown, NJ, USA, 1994. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/981732.981751.

R. Yan. *Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval*. PhD thesis, Canegie Mellon University, 2006.

R. Yan and A. G. Hauptmann. The combination limit in multimedia retrieval. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 339–342, New York, NY, USA, 2003. ACM. ISBN 1-58113-722-2. doi: http://doi.acm.org/10.1145/957013.957086.

R. Yan and A. 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. ISBN 1-59593-369-7. doi: http://doi.acm.org/10.1145/1148170.1148228.

R. Yan and A. G. Hauptmann. A review of text and image retrieval approaches for broadcast news video. *Information Retrieval*, 10(4-5): 445–484, October 2007. ISSN 1386-4564 (Print) 1573-7659 (Online). doi: 10.1007/s10791-007-9031-y. URL `http://www.springerlink.com/content/r742245481q23631/`.

R. Yan, J. Yang, and A. G. Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 548–555, New York, NY, USA, 2004. ACM. ISBN 1-58113-893-8. doi: http://doi.acm.org/10.1145/1027527.1027661.

J. Yang and A. G. Hauptmann. Exploring temporal consistency for video analysis and retrieval. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 33–42, New York, NY, USA, 2006. ACM. ISBN 1-59593-495-2. doi: http://doi.acm.org/10.1145/1178677.1178685.

J. Yang and A. G. Hauptmann. (un)reliability of video concept detection. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 85–94, New York, NY, USA, 2008a. ACM. ISBN 978-1-60558-070-8. doi: http://doi.acm.org/10.1145/1386352.1386367.

J. Yang and A. G. Hauptmann. A framework for classifier adaptation and its applications in concept detection. In *MIR '08: Proceeding of the 1st ACM international conference on Multimedia information retrieval*, pages 467–474, New York, NY, USA, 2008b. ACM. ISBN 978-1-60558-312-9. doi: http://doi.acm.org/10.1145/1460096.1460171.

E. Yilmaz and J. A. Aslam. Inferred ap : Estimating average precision with incomplete judgments. In *Fifteenth ACM International Conference on Information and Knowledge Management (CIKM)*, pages 102–111, New

York, NY, USA, November 2006. ACM. ISBN 1-59593-433-2. doi: http://doi.acm.org/10.1145/1183614.1183633.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: http://doi.acm.org/10.1145/383952.384019.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004. ISSN 1046-8188. doi: http://doi.acm.org/10.1145/984321.984322.

W. Zheng, J. Li, Z. Si, F. Lin, and B. Zhang. Using high-level semantic features in video retrieval. In *Image and Video Retrieval*, volume Volume 4071/2006, pages 370–379. Springer Berlin / Heidelberg, 2006. ISBN 978-3-540-36018-6. doi: 10.1007/11788034_38. URL `http://www.springerlink.com/content/dj1x764151607565/`.

Index

Abstract

This thesis considers concept-based multimedia retrieval, where documents are represented by the occurrence of concepts (also referred to as semantic concepts or high-level features). A concept can be thought of as a kind of label, which is attached to (parts of) the multimedia documents in which it occurs. Since concept-based document representations are user, language and modality independent, using them for retrieval has great potential for improving search performance. As collections quickly grow both in volume and size, manually labeling concept occurrences becomes infeasible and the so-called concept detectors are used to decide upon the occurrence of concepts in the documents automatically.

The following fundamental problems in concept-based retrieval are identified and addressed in this thesis. First, the concept detectors frequently make mistakes while detecting concepts. Second, it is difficult for users to formulate their queries since they are unfamiliar with the concept vocabulary, and setting weights for each concept requires knowledge of the collection. Third, for supporting retrieval of longer video segments, single concept occurrences are not sufficient to differentiate relevant from non-relevant documents and some notion of the importance of a concept in a segment is needed. Finally, since current detection techniques lack performance, it is important to be able to predict what search performance retrieval engines yield, if the detection performance improves.

The main contribution of this thesis is the uncertain document representation ranking framework (URR). Based on the Nobel prize winning Portfolio Selection Theory, the URR framework considers the distribution over all possible concept-based document representations of a document given the observed confidence scores of concept detectors. For a given score function, documents are ranked by the expected score plus an additional term of the variance of the score, which represents the risk attitude of the system.

User-friendly concept selection is achieved by re-using an annotated development collection. Each video shot of the development collection is transformed into a textual description which yields a collection of textual descriptions. This collection is then searched for a textual query which does not require the user's knowledge of the concept vocabulary. The ranking of the textual descriptions and the knowledge of the concept occurrences in the development collection allows a selection of useful concepts together with their weights.

The URR framework and the proposed concept selection method are used to derive a shot and a video segment retrieval framework. For shot retrieval, the probabilistic ranking framework for unobservable events is proposed. The framework re-uses the well-known probability of relevance score function from text retrieval. Because of the representation uncertainty, documents are ranked by their expected retrieval score given the confidence scores from the concept detectors.

For video segment retrieval, the uncertain concept language model is proposed for retrieving news items – a particular video segment type. A news item is modeled as a series of shots and represented by the frequency of each selected concept. Using the parallel between concept frequencies and term frequencies, a concept language model score function is derived from the language modelling framework. The concept language model score function is then used according to the URR framework and documents are ranked by the expected concept language score plus an additional term of the score's variance.

The Monte Carlo Simulation method is used to predict the behavior of current retrieval models under improved concept detector performance. First, a probabilistic model of concept detector output is defined as two Gaussian distributions, one for the shots in which the concept occurs and one for the shots in which it does not. Randomly generating concept detector scores for a collection with known concept occurrences and executing a search on the generated output estimates the expected search performance given the model's parameters. By modifying the model parameters, the detector performance can be improved and the future search performance can be predicted.

Experiments on several collections of the TRECVid evaluation benchmark showed that the URR framework often significantly improve the search performance compared to several state-of-the-art baselines. The simulation of concept detectors yields that today's video shot retrieval models will show an acceptable performance, once the detector performance is around 0.60 mean average precision. The simulation of video segment retrieval suggests, that this task is easier and will sooner be applicable to real-life applications.

Overview of Notation

	General
d	Document
Ω	Document Universe
$\mathcal{D}$	Collection $\mathcal{D} = \{d_1, ..., d_N\}$
Ω	Universe of documents
q	Query
$dom(\cdot)$	Domain of variable or feature
F	Feature $F : \Omega \rightarrow dom(F)$
$f(d)$	Feature value of feature f for document d
$\mathcal{V}$	Document feature vocabulary $\mathcal{V}$ a set of features
$\vec{F}$	Generic document representation for current query $\vec{F} = (F_1, ..., F_n)$ each $F_i \in \mathcal{V}$
$\vec{QF}$	Query features
w	Weighting function $w : \mathcal{V} \rightarrow \mathbb{R}$
$selectNweight_{ID}(\vec{qf})$	Selection and weighting method, results in $(\vec{F}, w)$
$retfunc_{ID}\langle\vec{F}, w\rangle(\vec{f} : dom(\vec{F}))$	Retrieval function instantiated for query representation $\vec{F}$ with weighting w accepting arguments of type $dom(\vec{F})$
$score_q(\vec{f})$	Score function for query q derived from a retrieval function, results in $\mathbb{R}$
b	Risk Attitude, system parameter
	Concrete features / Random variables
lf/LF	low-level feature $LF : \Omega \rightarrow \mathbb{R}$
o/O	Confidence score feature $O : \Omega \rightarrow \mathbb{R}$
t/T	Term occurrence feature $T : \Omega \rightarrow \mathbb{B}$
tf/TF	Term frequency feature $TF : \Omega \rightarrow \mathbb{N}$
c/C	Concept occurrence feature $C : \Omega \rightarrow \mathbb{B}$
cf/CF	Concept frequency feature $CF : \Omega \rightarrow \mathbb{N}$
r/R	Relevance to the current query $R : \Omega \rightarrow \mathbb{B}$
	Concrete weights
$P(C\|R)$	Probability of concept occurrence
	Collection Statistics
$P(C)$	Concept Prior