An Investigation Into Weighted Data Fusion for Content-Based Multimedia Information Retrieval

Peter Wilkins

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

DCU

Dublin City University

School of Computing

Supervisor: Prof. Alan F. Smeaton

6th July, 2009

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work

Signed:

ID No:

Date:

Abstract

Content Based Multimedia Information Retrieval (CBMIR) is characterised by the combination of noisy sources of information which, in unison, are able to achieve strong performance. In this thesis we focus on the combination of ranked results from the independent retrieval experts which comprise a CBMIR system through linearly weighted data fusion. The independent retrieval experts are low-level multimedia features, each of which contains an indexing function and ranking algorithm. This thesis is comprised of two halves. In the first half, we perform a rigorous empirical investigation into the factors which impact upon performance in linearly weighted data fusion. In the second half, we leverage these finding to create a new class of weight generation algorithms for data fusion which are capable of determining weights at *query-time*, such that the weights are *topic dependent*.

Acknowledgements

The writing of the acknowledgements is something dreamt about as I got ever closer to finishing. However now that it is 11.45pm on a night dangerously close to the end of the academic year, it's very tempting to say so long and thanks for the fish. Of course the last minute nature of this fits in fairly well with my general experience of the thesis, and deadlines in general.

I would like to thank my supervisor Alan Smeaton for taking me on and giving me the freedom to explore different research directions, whilst getting me involved in various multi-institute activities. This certainly has given me great experience, a broad perspective, and presented opportunities I would not have otherwise enjoyed. Likewise I would like to thank Ross Wilkinson for taking me on as an intern and software engineer many years ago, introducing me to research back in Melbourne.

Having been in Ireland for over six years, I often get asked how did I handle the move. My answer is that i've been incredibly lucky to have ended up in the CDVP because of the great group of people, past and present, which make up its members. Whilst we certainly do good work, there's no denying that the social element is beyond what you find elsewhere. The camaraderie and the craic make it a place where you want to come in, because you'll be hanging out with great people, and that in turn creates the environment where we end up doing good stuff. To everyone in the CDVP, my unreserved thanks.

Typically at this point, the traditional shout-out occurs. This has been mathematically proven to be a complete rank ordering of your friendships, where those first are closest confidants clearly in line for getting something in a will, or at least from your desk when you move on, and those last probably owe you some money (you know who you are). To attempt to break this cycle, everyone of the following has been a good friend and has helped me through the thesis process. To add some ambiguity over the ordering, nicknames will be used. N.b. some may not be aware of their nicknames, but creative thinking should sort it out. My many thanks to; Fergie, C-Dog, Master J, Bouncy McBouncy, GG, Rabbit, K^2, Sir Burnsalot, McChug, Gerry Xiaoping, Sensei Lee, Macca, Tom, Mr. Fabulous, Rosie, Clapton and Hock. If you didn't see your name in there, think harder. I've also been lucky to have gotten support from back in Australia, through my friends Joe & Em and the rest of the scoobies, and Rowan & Kim for demonstrating what life in a research lab should be like.

The love, support and patience of a family is great to have, and I would like to give my sincerest thanks to Mum, Dad and Simon. The endless encouragement and support has always been a fantastic help and is so very much appreciated. Similarly I would also like to thank Mike & Mary Hearne, for the encouragement and freedom whilst writing up down in Kilkenny.

Finally I would like to give my heartfelt thanks to my girl Mary, who has certainly been a PhD widow in recent years. Her experience of having been through it all before was invaluable to me in helping to write and refine my ideas and experiments. The occasional butting of heads, and when required, cop-on sentiment, meant I was never left wanting for a contact sport replacement either! In all seriousness though, the love, support and understanding she gave through the years, provided me with the confidence to finish, and greatly advanced my thinking like a researcher. Mary will also be happy to know that i've finally learned how to spell retreival.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Content-Based Multimedia Information Retrieval (CBMIR) systems are complex retrieval platforms which leverage multiple areas of research from signal processing and machine learning through to relevance feedback and classification. Multimedia data is inherently noisy by nature, often as a result of the capture of an event and its translation to a digital format introducing some form of signal loss. One of the fundamental characteristics of any of the approaches to multimedia retrieval is to utilise multiple noisy sources of data and combine them in such a way that the performance of the end system is greater than the sum of its parts, a point which this thesis will reinforce.

The combination of various forms of data to answer an information need for a given query has generated many different types of solutions to the CBMIR problem. One area of active research, for instance, is that of high-level semantic concept detection, where the noisy signals are classified into an ontology of concepts which can then be exploited for retrieval. Our focus in this thesis however is to treat each of these separate sources of information as an independent retrieval authority, otherwise known as a retrieval expert. The CBMIR systems we build make use of multiple retrieval experts which are independently queried with each query component, to generate a ranked lists of documents for combination into a final result. This combination – the merging of result lists from separate retrieval systems – is known as *data fusion*. As we have stated, multimedia data is inherently noisy, for certain topics, certain retrieval experts will perform better than others. To maximise performance weighting schemes are employed to effectively combine these ranked lists and promote those which offer more relevant results. This thesis studies weighted data fusion and its application to CBMIR.

1.1 Motivation

The determination of the weights to employ for data fusion with CBMIR is a non-trivial research problem which has seen numerous approaches developed. Fundamentally however, the application of weights to the ranked results from retrieval experts is only part of the problem to be addressed. For ranked lists to be effectively combined their scores require normalisation such that raw values from differing lists do not saturate any combination function. How should normalisation occur? Should it occur based upon document scores or based upon document ranks? Likewise, how many results from each expert should be read? With the use of multi-part queries, where experts return multiple ranked lists for a single topic, how should these results be aggregated? Hierarchically, for example? And, if so, what form of hierarchy should be used? Many combination operators have been previously developed, such as CombSUM and CombMNZ (Fox and Shaw, 1994). Which is the most effective when used with weighting? Finally, to the weights themselves: is there any commonality to weight distributions or is their form collection-dependent? And, in any event, can this information be exploited?

This set of incomplete questions demonstrates the sheer number of variables which come into play when weighted data fusion is employed within the domain of CBMIR. Many of these factors may be considered as engineering details, or may be thought to provide only a minor impact upon the final result for a query. We believe, however, that the impact of these factors is underestimated, and that they can play a far more significant role in retrieval performance than previously thought.

As such this thesis is comprised of two halves, in the first of which we conduct a thorough examination of weighted data fusion and all of its variables. This allows us to determine which variables impact upon performance, whether experimental data from other domains such as text retrieval is consistent with our outcomes, and what level of performance can we expect an optimised data fusion scheme to provide. In The second half of this thesis we present a review of weight generation schemes for data fusion and present novel algorithms which experimental knowledge presented in the first part of this thesis allowed us to develop.

We believe this work is of crucial performance as the development of data fusion algorithms cannot occur within a vacuum. This study is in part designed to determine what is the maximum performance that can be achieved with data fusion, such that we obtain an upper bound on empirical performance. Without knowing this, the evaluation of data fusion algorithms is constricted to measuring improvement against other approaches, rather than against a global maximum which would inform us as to how effective our algorithms really are.

This thesis is thus driven by two key research objectives:

1. We believe that the complex nature of weighted data fusion within CBMIR involves the interplay of a significant number of factors, each of which can potentially impact upon retrieval performance. Our objective is to conduct a rigorous empirical evaluation of these factors so as to determine their importance within data fusion schemes. We achieve this through a study of the ideal topic-level weights, thereby determining weighting attributes which weight generation algorithms for data fusion should seek to emulate.

2. Our second objective is to apply the findings of this examination to produce a novel weighting scheme for data fusion which leverages our observations for improved retrieval performance, offering capabilities which we believe are not matched by existing algorithms.

1.2 Thesis structure

In this section we describe the layout of this thesis, with an overview of the chapters comprising it. The thesis is comprised of five main content chapters, followed by a concluding chapter revisiting the major outcomes of this research.

Chapter 2 In Chapter 2 we provide an overview of multimedia information retrieval, the impact of multimedia data on the retrieval process and work which aims to leverage multimedia data to aid retrieval. Multimedia information retrieval is a wide area of research which incorporates techniques from many sub-fields of research from signal processing through to machine learning. This chapter provides context for where our research sits within this wider research domain. We cover the properties of multimedia data and how these create noisy sources of data compared to authored text, the representation of multimedia data, alternative methods for resolving the ambiguities of multimedia data and finally the evaluation metrics we use in this thesis.

Chapter 3 As we've highlighted, CBMIR systems are highly complicated as they incorporate many forms of evidence in order to arrive at a response to a query. The task of our CBMIR system is to combine multiple ranked lists of results from multiple low-level retrieval experts through linearly weighted data fusion. In this chapter, we present an explanation of data fusion and previous research into its study. The application of data fusion involves the use of several algorithms or variables, each of which can have a demonstrable impact upon retrieval performance. We identify each of these methods or variables and explain their application and purpose. Whilst several of these variables have been studied before, we believe that this is the most thorough examination of both explicit and implicit variables and methods which can affect weighted data fusion performance.

Chapter 4 Chapter 4 presents a methodical, rigorous examination of the factors which impact upon weighted data fusion performance identified in Chapter 3. The significance of our examination compared to previous studies lies in the use of an optimisation technique known as *coordinate ascent*. This method allows us to determine, for any given set of inputs, what the near-ideal linear set of weights for combination are. We deviate from the standard experimental model by performing these optimisations directly on the test data. In this chapter we provide a full explanation and justification for this approach; our primary motivation is that it allows for direct study of the ideal weights to determine if there are properties of their distribution which should be emulated. Our examination is thorough, and we contrast our evaluation with to the early data fusion experiments of Lee (Lee, 1997) to review our findings against previous experimental conclusions.

Chapter 5 Having established in Chapter 4 the ideal distribution of weights for data fusion, and the variables which have the most impact upon retrieval performance, in chapter 5 we present a review of related work on weighted data fusion and the approaches which can be used for the creation of weights. This chapter reviews approaches such as query-independent weighting, query-class research and machine learning approaches. We find that whilst many of these approaches offer several advantages and achieve good performance, none of the defined approaches satisfy all the criteria we establish in Chapter 4 as necessary for achieving optimal performance with data fusion.

Chapter 6 In this chapter, we define our own set of algorithms for creating the weights to be used in linear weighted data fusion. We first define the motivation for our proposed approach, so as to establish where it fits within the 'family' of weighted data fusion algorithms. Second, we provide an overview of some of the characteristics of ranked result lists which we aim to exploit. Third, we present our novel algorithms for generating linear weights, followed by experiments on the corpora we have utilised throughout this thesis. Finally, we analyse our results, and attempt to determine why aspects of our algorithms work.

Conclusions The conclusions chapter summarizes the outcomes of each of the chapters within the thesis. Following this we provided a brief reflection on these outcomes, and highlight our perspective on CBMIR and current research in the field with emphasis on how the work of this thesis compliments the area.

Chapter 2

Overview of Content-Based Multimedia Information Retrieval

In this chapter we give an overview of multimedia information retrieval, the impact multimedia data has on the retrieval process and research which aims to leverage multimedia data to aid retrieval. The field of multimedia retrieval is a very broad research area, with many disparate approaches falling under its banner. Our aim in this chapter is to provide some context of the relevance of our work in investigating weighted data fusion for multimedia retrieval and how it fits into the general multimedia research domain.

Research into Information Retrieval (IR) incorporates the representation, storage, organisation and the access of information. The key task of IR is given a stated information need, the IR system is to return useful information for that need (Baeza-Yates and Ribeiro-Neto, 1999). This is distinct from returning *data* for an information need, such as a database query, where the stated request is explicit, the data is stored and organised in a relational form, and what is returned must be a precise match. IR systems therefore operate with a degree of uncertainty, which is resultant from the many components which comprise an IR system, such as how a natural language document is represented within the system, the translation of an information need into a form which the IR system can interpret and the presentation of the final result for a request (Baeza-Yates and Ribeiro-Neto, 1999).

Multimedia Information Retrieval (MIR) builds upon IR research by incorporating forms of information which are not restricted to textual sources of data. Typically when we refer to IR we implicitly mean IR with regards to natural language documents. MIR on the other hand can refer a multitude of data types including audio, video and visual sources of data. The use of the term MIR implies that we are studying some aspect of IR research utilising at least one form of data which is non-textual (Blanken et al., 2007).

The evolution of MIR research has in many ways mirrored the evolution of text based IR research. Early IR systems evolved from the use of libraries, where a library patron would describe to a librarian what information they were trying to find. The librarian would utilise expert knowledge and resources such as card indexes, which described for each book its authors, title, potentially a summary or key terms describing the content, and the books location. Early IR systems could be considered as *metadata* retrieval systems, where a search was conducted within manually constructed data such as that available in card catalogues, and the search was against information which described the books rather than the books themselves (Baeza-Yates and Ribeiro-Neto, 1999). Likewise early experimental MIR systems utilised metadata to retrieve non-text data, such as image retrieval techniques which utilised the captions assigned to an image allowing for keyword based retrieval (Smeaton and Quigley, 1996). Just as text based IR systems evolved to incorporate the *content* of the documents it was searching, so to do MIR systems.

Content-Based Multimedia Information Retrieval (CBMIR) therefore is concerned with the representation, storage and retrieval of multimedia data at a content level. For visual data this may be the colour histogram of an image which is stored and retrieved against, whilst for audio this may be a temporal segment which has undergone Automatic Speech Recognition (ASR) and is then indexed. This chapter will provide an overview of CBMIR, including how data is represented, how we retrieve it and how we evaluate what is retrieved. In the next section we will review some of the properties of multimedia data which highlight the challenges it presents for CBMIR.

2.1 Properties of Multimedia Data

Multimedia data has several properties which impact upon its performance in retrieval tasks, especially when compared to that of text data. Two of these key properties which effect retrieval are known as the *Sensory Gap* and the *Semantic Gap* which combine to make the MIR task more difficult. In summary, the sensory gap concerns the introduction of noise when multimedia content is generated, whilst the semantic gap concerns the difference between how a retrieval system interprets a document and how a human may interpret that document.

2.1.1 The Sensory Gap

One of the interesting properties of multimedia data is that often it is a digital capture of some natural scene or event. Whilst text documents are now typically authored digitally, many multimedia sources undergo an Analogue to Digital Conversion (ADC) process. This digitisation has allowed very large quantities of multimedia data to become exploitable by information systems, but also is inherently coupled with a degree of additional noise which can impact upon the quality of any system operating on that data. Smeulders et al. (2000) characterises this for visual data as the 'sensory gap':

> "The sensory gap is the gap between the object in the world and the information in a (computational) description derived from a recording of that scene" (Smeulders et al., 2000).

This observation is generally applicable to most forms of multimedia data which require some form of capture. Smeulders *et al.* elaborate with regards to visual data that the sensory gap introduces a degree of uncertainty about what is being captured, factors such as the positioning of a camera, the illumination of the scene, is it a 2D recording of a 3D or 2D scene etc, can all impact upon the quality of the signal and how it should be treated by a retrieval system.

Coupled with noise which may be introduced at capture time, raw multimedia signals require very large amounts of data to store. For instance, taking a 800 × 600 image stored with colour data, where the colour is represented as Red, Green and Blue (RGB) of one byte each, the resulting uncompressed image requires 1.44Mb of storage space (Blanken et al., 2007). The situation is worse for digital video, where storing a colour video of 25 frames per second (fps) with a duration 90 minutes requires 112 Gigabytes (Smeaton, 2004). Because of this compression of multimedia data is required, with popular standards such as MPEG-1 and JPG in common use for video and still images (Manjunath et al., 2002). All of these compression standards however are 'lossy', and as such after compression we lose some of the information that was present in the original representation. Whilst this can further contribute to noise present in multimedia signals, the benefits of compression have made ubiquitous access to multimedia content possible today.

As a comparison of different sources of data, the .GOV document collection from TREC had a total size of 18.1 Gb and contains 1.25 million text documents. As a comparison, 18 Gb of TRECVID MPEG-1 digital video provides us with approximately 50 hours of content. Detailed later in the chapter, when we handle video data we typically decompose the video into 'shots', which are small visually homogeneous segments of video at least two seconds in duration, and which become our unit of retrieval for video retrieval tasks. Therefore for equivalently sized collections, we have from the text collection a corpora of 1.25 million documents in which to experiment, whereas for digital video we have approximately 90,000 retrieval units. This poses challenges for the experimentation and evaluation of CBMIR systems, particularly within digital video, as we are required to operate with comparatively smaller corpora in terms of retrieval units.

One may now think that because of the reduced size in the corpora of retrieval units, that retrieval experiments should perhaps be easier than text equivalents. A fundamental problem however is the representation of multimedia data typically provides no clues as to what the content may be about. Whilst text data certainly has research issues with the interpretation of documents, particularly in areas such as disambiguation or sentiment analysis, if a text document contains the keywords 'kitten' and 'cat' multiple times, it is likely to have something to do with cats, whereas a multimedia document may have a feature vector of '0 4 3 7 1'. This problem of interpretation of low-level multimedia data is often referred to as the 'semantic gap'.

2.1.2 Semantic Gap

The semantic gap was explicitly defined by Smeulders et al. (2000) to address the disappointment felt by the performance of early content-based image retrieval approaches, and has achieved near ubiquitous use in multimedia publications. Smeulders *et al.* define the semantic gap within the context of visual data:

> "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" (Smeulders et al., 2000).

As with the previous sensory gap, the applications of the semantic gap can be widely applied to other forms of content-based multimedia data, not just static images. Effectively the use of the term semantic gap refers to the discrepancy between the system's representation of multimedia data and how that system may relate and process the data, compared to how an end user perceives that data. For instance, if we have two images for which the system extracts visual features and finds both have a large blue area, a yellow circle and a green block, then the system would consider these two images to be very similar. An end user examining these images however would see one was a scene on a sunny day in a park, and the other a sports shot of tennis being played.

At a more general level the semantic gap can be considered to be a specialisation of the interpretation of *relevance*, which is the fundamental cornerstone of any IR system (Van Rijsbergen, 1979). Indeed Smeulders *et al.* in their definition of the semantic gap make reference to the interpretation of the data being situation specific, one of the potential types of relevance, along with algorithmic relevance, topical relevance, cognitive relevance and affective relevance (Saracevic, 2007). A detailed discourse on relevance is beyond the scope of this work, however the comparison to relevance research highlights the research difficulty of the task of a retrieval system inferring some form of semantics onto raw multimedia data. Given that we know there are many types of relevance for a given retrieval scenario, equally there are likely to be many semantic interpretations of multimedia content dependent on the multimedia retrieval scenario. As with relevance however multiple branches of multimedia research attempt to tackle the problem, notably areas of relevance feedback and high-level semantic concept detection (Datta et al., 2005). Later in this chapter we will briefly examine the area of semantic concept detection.

Proponents of relevance feedback highlight its ability to help overcome the semantic gap as the inclusion of a 'human in the loop' enables a system to better interpret the information need and provide more relevant results (Zhou and Huang, 2003). There is no doubt that the inclusion of direct feedback from a user regarding an information need will lead to demonstrable improvements in retrieval performance. As stated previously the focus of our thesis is on weighted data fusion, which would occur prior or in conjunction with relevance feedback, and as such the two approaches can be considered as complimentary. We conduct a review of relevance feedback approaches for CBMIR in section 5.6.1.

For any CBMIR system to interact with multimedia data, the data requires some form of representation. In the following section we will briefly review representations of multimedia data and their application to CBMIR systems.

2.2 Representing Multimedia Data

For multimedia content to be exploitable by a CBMIR system the data needs to be transformed into an appropriate format. In this section we will briefly examine standards such as MPEG-7 which are designed to represent multimedia content to allow interoperability between systems, low-level feature extraction which takes a raw multimedia signal and produces some form of data, and finally high-level semantic concepts which attempts to infer some meaning or interpretation to multimedia data. Broadly speaking, MPEG-7 is a markup standard that can encompass both low-level features and semantic concepts. Low-level feature extraction refers to techniques and algorithms which process multimedia data using unsupervised methods. Finally semantic concept detection is typically supervised a approach which utilise prior training data or domain knowledge.

2.2.1 Metadata Representations

Formally known as the Multimedia Content Description Interface, MPEG-7 is an all encompassing standard whose focus is on the description of multimedia content (Chang et al., 2001; Manjunath et al., 2002). In terms of multimedia it is able to handle multiple modalities including image, video, audio, speech, graphics and combinations of these. MPEG-7 allows for multimedia content to be described not only temporally but spatially as well. MPEG-7 is comprised of descriptors, descriptor schemes and a description definition language. Descriptors define the syntax and semantics of audio-visual content, covering attributes such as colour, motion and energy. Implementations of descriptors are what we use in our work for our visual retrieval experts, which we define later in Section 3.3.2. Description Schemes allow for specifying the semantics of relationship between descriptors and other description schemes. Finally the Description Definition Language allows for the flexible specification of descriptors and description schemes based on XML schemas. This flexibility and the general size of MPEG-7 whilst giving it great expressive power, can make its use at times cumbersome or ambiguous. To counter this, research groups such as Joanneum Research (JRS) have developed a Detailed AudioVisual Profile (DAVP) to provide more structure to the standard (Bailer and Schallauer, 2006). As part of the consortium which developed MPEG-7, a reference implementation known as the MPEG-7 experimentation software was built, which members of our research group have extended and implemented to provide access to MPEG-7 features (O'Connor et al., 2005). Further details on MPEG-7 can be found in the works of Chang et al. (2001) and Manjunath et al. (2002).

2.2.2 Low-Level Features

The extraction of low-level features from multimedia data is analogous to the extraction of terms from text documents and can be considered as part of an indexing phase in a CBMIR system. Low-level feature extraction is typically a fully automatic process. We use the phrase 'low-level' to describe these features as they impart no semantic or higher level understanding of the data, but rather they output either data patterns or statistics of the data which is being analysed. Low-level features are the foundation of most CBMIR systems upon which either more advanced features can be built or retrieval systems constructed from. Their independent use for retrieval however, particularly ad-hoc retrieval in unconstrained domains illicit poor performance (Smeulders et al., 2000). In general there are three major types of multimedia data, images, audio and video. Images are a static form of data, whilst both audio and video have a temporal component.

Images can be processed for several low-level features, notably for *colour*, *texture* and *shape*. Colour is one of the most widely used features in visual processing and one of the most effective. We can compute for an image the distributions of its colour and represent these as histograms. These can be averaged across an entire image, or the image can be divided into sub-regions where the averages can be computed for each. Representations of colour are dependent upon the colour space, which provides a model for representing colours as numbers. Common colour spaces such as RGB are hardware oriented, tailored for displaying images on monitors, whilst colour spaces such as HSV (Hue, Saturation, Value) are based more on how colours are perceived. The text by Humphreys and Bruce (1989) provides a good discussion of how we perceive colour and the impacts this can have. Texture based features examine patterns which occur within an image, as opposed to colour where each pixel can be considered independent. Texture features can determine if there are dominant orientations or patterns in an image and can be very useful for certain classes of query. Shape based features are able to capture local geometric regions within an image. An overview of visual features can be found in Rui et al. (1999); Smeulders et al. (2000); Datta et al. (2005, 2008), whilst the visual features we will be utilising are detailed in Section 3.3.2.

Audio features can be extracted from amplitude-time sequences, where we can detect attributes such as *periods of silence*, the *average energy of a signal or the zero crossing rate (ZCR)* which indicates how often the sign of the amplitude changes (Blanken et al., 2007). In our work, we will not deal with low-level audio features directly, however we will be utilising text derived from automatic speech recognition (ASR) algorithms. Whilst ASR approaches themselves may not be characterised as a low-level feature, as many approaches require some form of a model to be derived in order to operate, the output from ASR is often noisy. Considering that the audio signal may often not be ‘clean’, that is that apart from a speaker talking there may be background noise or additional speakers, so high accuracy cannot be expected. As such in our work we treat the text output of ASR as a low-level feature as like other low-level features it is not noise free.

Video is a sequence of moving images, displayed at a rate of at least 25 Frames Per Second (fps) to provide the illusion of motion (Smeaton, 2004). A multi-modal form of data itself, it incorporates both visual and audio features. Our previously described low-level features for images and audio can be utilised in the analysis of digital video. The fundamental difference between images and video is the temporal nature of video. In order to apply many of the low-level image features to video we are required to extract frames from the video and treat each of these as a separate images. The question when dealing with video is at what sample rate or method should we extract the keyframes. One approach is to select a uniform time, and consistently sample at that interval, for example extracting an image every five seconds. An alternative approach is to determine what structure exists within the video and sample from these identified units. One common unit used is that of a 'shot'. A shot, as defined by TRECVID, is a visually homogeneous segment of video which is of at least two seconds duration. Shot boundaries can be computed automatically from video, a review of which is provided by Smeaton et al. (2009). An alternative segmentation is to utilise a more semantic unit from which to sample, this however is dependent upon the corpus of video under consideration and requires uniformity within that corpus. One example is to use 'story bounds' in broadcast news video (Kraaij et al., 2004), where in this context, the content within the story bound was all semantically related to the news story it was representing.

As stated previously, the majority of low-level features are unsupervised approaches which require no human intervention or effort. If human effort is available, such as to conduct annotation activities, then more complex feature extraction activities become available, notably for multimedia retrieval the application of High-Level Semantic Concept Detection.

2.2.3 High-Level Semantic Concepts

High-Level Semantic Concepts are a rapidly expanding area of multimedia research which aims to bring some form of interpretation to multimedia data to allow for easier querying. In the general sense High-Level Semantic Concepts (henceforth referred to as 'concepts'), take a multimedia signal, and apply to it some label which represents the content of the multimedia signal. For instance, these methods allow us to assign labels such as 'outdoor', 'sky', 'person', 'face', etc, to multimedia content, so that an end user can specify the keywords 'person outdoor' and should have returned for that query content of people in an outdoors setting. As such, they offer a potential avenue for helping to bridge, or at least reduce, the semantic gap as noted by Hauptmann:

> “this (High-Level Semantic Concepts) splits the semantic gap between low-level features and user information needs into two, hopefully smaller gaps: (a) mapping the low-level features into the intermediate semantic concepts and (b) mapping these concepts into user needs” (Hauptmann, 2005).

The general requirements for concept detection is a corpus of video split into training and test collections, a set (or potentially an ontology) of concepts which occur in that video and a set of annotations which define both positive and negative occurrences of the concepts in the training data. Early work in concept detection was focused on domain specific applications, such as that of Zhang et al. (1995), where the domain was news video, and the shots were classified into news shots or anchor-person shots. Other examples of concept detection within fixed domains include the detection of advertisements from broadcast television (Sadlier et al., 2002), and the detection of significant events occurring within sports videos such as scores, fouls, etc, (Babaguchi et al., 2002; Sadlier et al., 2003).

Concept detection has evolved into a multi-modal process, which incorporates evidence from visual, audio and text based modalities, whilst leveraging research developed in computer vision and machine learning communities. The emphasis on concept detection now is the move to more generalised semantic indexing, rather than utilising specific cues within the multimedia data which correlate with semantic events which are known because of domain knowledge. Concept detection has proven to be a key component of digital video retrieval systems, and the TRECVID benchmarking workshops have proven crucial to the development of generic approaches to semantic indexing. This was achieved by requiring groups who participate in the semantic concept detection activity of TRECVID to submit results for all of the concepts specified (typically between 20-30 concepts), rather than implementing a handful of detectors with hand crafted heuristics.

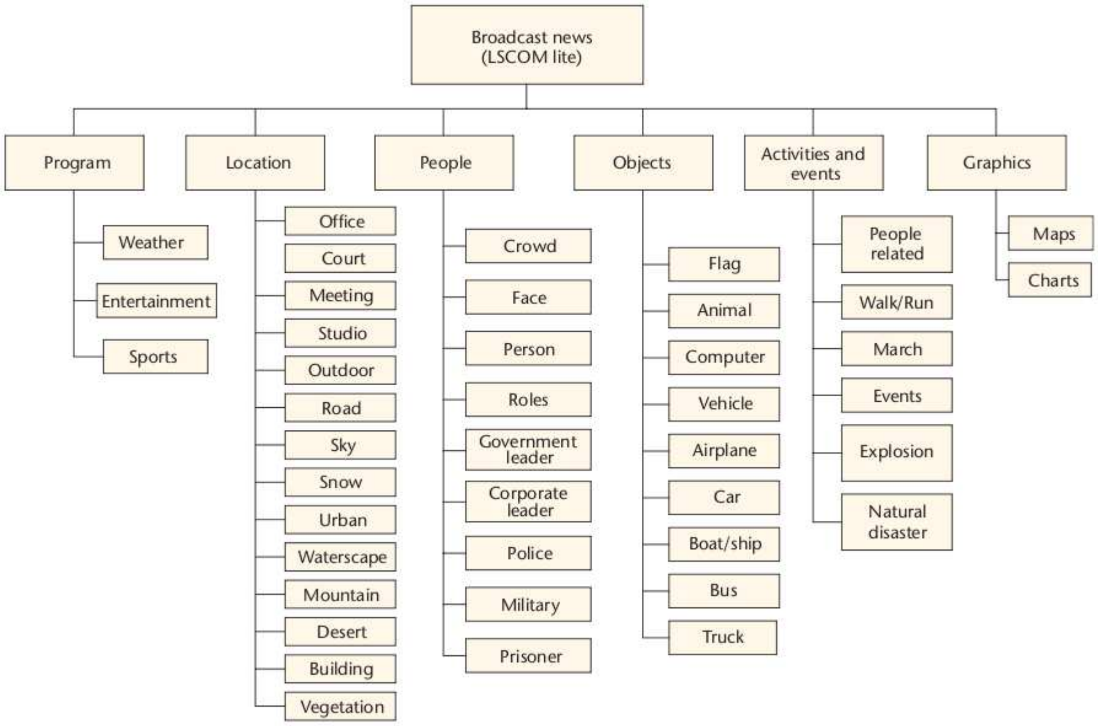


Figure 2.1: LSCOM-Lite ontology as used in TRECVID 2005 and 2006 (Naphade et al., 2006).

One of the primary challenges for concept detection is the specification of what are the semantic concepts that should be detected. An early initiative in this area was the development of the LSCOM annotations (Large Scale Concept Ontology for Multimedia) (Naphade et al., 2006). This activity was designed to develop a taxonomy of 1,000 semantic concepts for describing broadcast video, developed in conjunction with multimedia researchers and domain experts. Of these, 449 concepts had annotations created for them from the TRECVID 2005 development data collection, resulting in 61,901 annotations per concept. A reduced subset of these concepts, known as LSCOM-Lite was used for the concept detection tasks in TRECVID 2005 & 2006 and are shown in Figure 2.1. This extensive annotation activity subsequently allowed the development of large scale generalised semantic concept frameworks.

Notable amongst these are the MediaMill 101 set (Snoek et al., 2006), the Columbia374 semantic detectors (Yanagawa et al., 2007) and the Vireo 374 set

(Jiang et al., 2007). The MediaMill system captures the influence of the production style in the creation of processed video, recognising that the creation of video is the result of an authoring process. Their system incorporates content analysis, style analysis and context analysis in an iterative learning framework. The Columbia system takes an alternate approach, opting for greater breadth of coverage in its semantic concept detectors by creating lightweight classifiers which utilised three visual features, which allowed for a greater number of detectors to be constructed. The Vireo 374 set takes a similar approach to Columbia, but integrates visual words and keypoint features into their classifiers. All of the aforementioned concept detectors achieve good performance on the concept detection task in TRECVID and have been shown to assist in the retrieval process. Whilst the utilisation of semantic concept detectors has demonstrable performance benefits for retrieval, their application is not without challenges.

The sensitivity of the models trained from one type of corpus when tested on another is of concern, and is an area of active research. Examining the results of the transition of corpus in TRECVID 2007 from 2006, models which were trained using 2006 training data performed poorly on 2007 data (Over et al., 2007), whilst a majority of groups utilised newly created annotations on the 2007 training data rather than reusing the 2006 annotations. Recent work by Jiang et al. (2008) addresses the cross domain issue by specifically examining the application of models trained on one particular corpus of data, tested with a corpus from a different domain. Efforts such as these are required, because the cost of completing annotation exercises consumes considerable resources, it is far more desirable to reuse existing annotations then having to generate new annotations every time a corpus changes.

More fundamental issues are raised by both Zhou and Huang (2003) and Santini and Dumitrescu (2008). Both sets of authors make fundamentally the same claim, that different users at different times will have different interpretations of what a semantic label may mean. In particular Santini and Dumitrescu argue that the interpretation of a document is context-dependent rather than its meaning being an independent property of the document. This argument is quite similar to that of information scientists who disagree with the use of static relevance judgements for the evaluation of information retrieval systems, arguing that relevance has various properties which make it dependent upon the context and situation of the user at the time of the query. The counter-claim from the perspective of laboratory IR experiments is that the use of fixed test collections and relevance assessments has allowed rigorous cross comparison of retrieval algorithms and a demonstrable improvement in their performance (Saracevic, 2007). In a similar fashion the application of semantic concept detectors, particularly in the case of digital video retrieval, has been shown to offer significant performance gains, however the study of these detectors is outside the scope of this work.

Our work however in this thesis is concentrating on the application of linearly weighted data fusion to noisy data sources to determine what factors, such as features and combinatorial algorithms, impact upon performance, and what performance can be achieved with what methods to successfully combine results. Whilst there is no doubt the use of semantic concepts will boost retrieval performance, the development of robust classifiers of wide semantic coverage is in itself an area of massive research, whose application would make the study of data fusion more problematic as we would need to disambiguate where performance influences originate whilst having to ensure that the classifiers we utilised were optimised to achieve good performance. As such we will not utilise them in this study.

The use of semantic concept detectors can compliment systems which utilize low-level features and data fusion and can be utilized in numerous ways. Firstly, concept detectors can be employed in the retrieval process to handle more generic information seeking tasks from a user, whereas QBE driven data fusion approaches require some concrete examples of the information need to start a query. For instance, a user looking for people outdoors may find it easy to examine the results of an intersection of a person and outdoor detector results. In this case data fusion could be employed to generate a ranking on this subset of images. Similarly a user may use both approaches in the formulation of a query, where they have concrete examples of what they are trying to find, and the system utilizes relevant concept detectors to reduce the search space in which to rank. These examples are simple cases of how these techniques can interact, so as to demonstrate the compatibility of a system employing both approaches to addressing the retrieval problem. Indeed many system participating in TRECVID would utilize both of these approaches (Snoek et al., 2005)(Smeaton et al., 2008).

2.3 Our CBMIR Environment

In this section we will present a high level overview of the organisation of our CBMIR system, the resources it will utilise and the types of operations it can undertake. The system we will be utilising in this thesis can be considered as an *experimental* system as our experiments are being conducted within the confines of a laboratory IR experiment (Van Rijsbergen, 1979). This system is designed to handle *ad-hoc* search tasks. An ad-hoc search task is where the documents we have indexed within the system remain static, whilst new queries are introduced (Baeza-Yates and Ribeiro-Neto, 1999). Whilst components of this system have previously been utilised for interactive retrieval experiments involving controlled groups of users, our investigation will only be conducted by operating as a *fully automatic* search system. Figure 2.2 provides an overview of the different types of operation that a search system can engage in, the other two variants being interactive and manual search. In our experimentation there will be no humans in the loop, as such queries will be processed as defined by our experimental corpora with no intervention. This allows for a robust examination of the retrieval experiments as there should be no variation in the queries any one particular algorithm will handle.

Smeulders et al. (2000) defines content-based search systems as either handling narrow or broad based domains. Narrow domains exist where the content to be indexed is relatively homogeneous, such as images of aircraft, which in that environ-

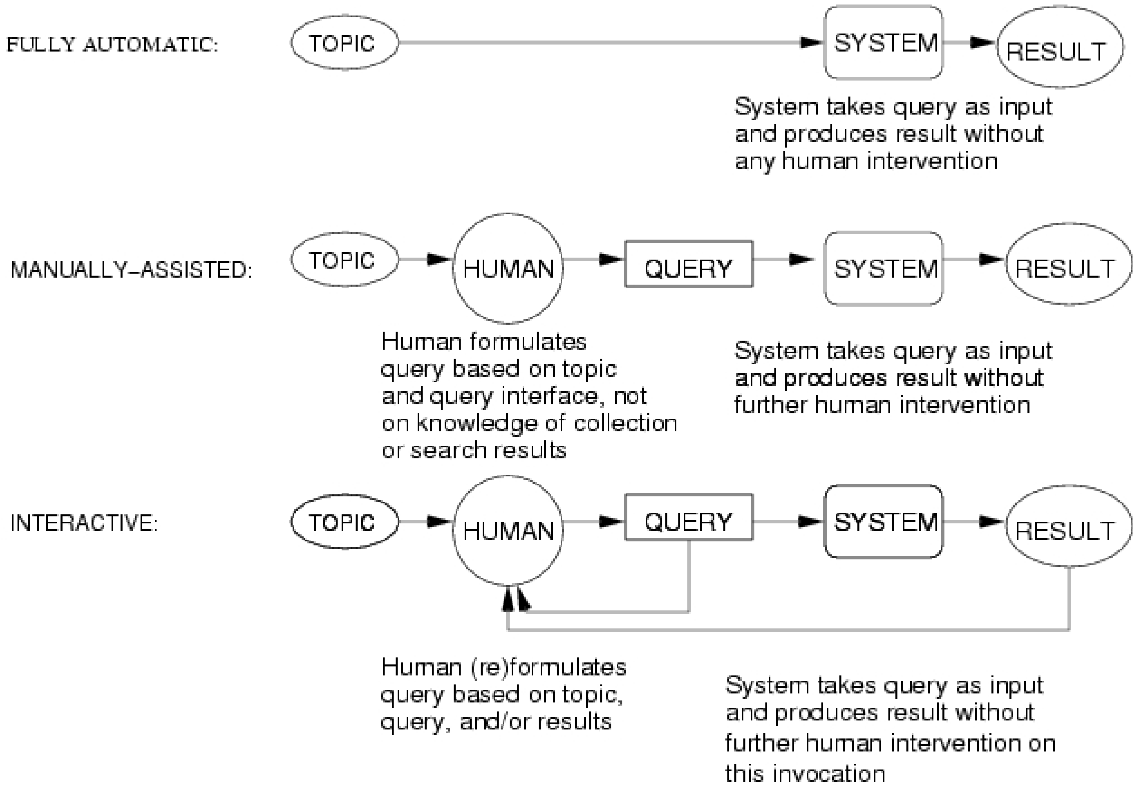

FULLY AUTOMATIC:
TOPIC
SYSTEM
RESULT
System takes query as input and produces result without any human intervention
MANUALLY–ASSISTED:
TOPIC
HUMAN
QUERY
SYSTEM
RESULT
Human formulates query based on topic and query interface, not on knowledge of collection or search results
System takes query as input and produces result without further human intervention
INTERACTIVE:
TOPIC
HUMAN
QUERY
SYSTEM
RESULT
Human (re)formulates query based on topic, query, and/or results
System takes query as input and produces result without further human intervention on this invocation

Figure 2.2: Query models for search (Smeaton et al., 2006)

ment heuristics can be crafted to exploit domain knowledge of the underlying collection. Our retrieval system can be considered as handling broad domains, where we handle data from both digital video and image collections from a variety of sources. Furthermore the retrieval algorithms we will evaluate and develop will make no use of any domain specific information in order to aid performance.

The retrieval resources we will be using in this thesis will be comprised of low-level features. We are not making use of any semantic concepts or domain knowledge in our experiments. For the purposes of this experiment, we consider text evidence to be a form of a low-level feature. Each of these retrieval resources has associated with it an index and a ranking function, making each low-level feature a complete independent search engine. Henceforth we refer to the combination of low-level features and a ranking function as a *retrieval expert*, where each expert is independent of the other experts utilised.

The queries which we will investigate will be multi-modal queries consisting

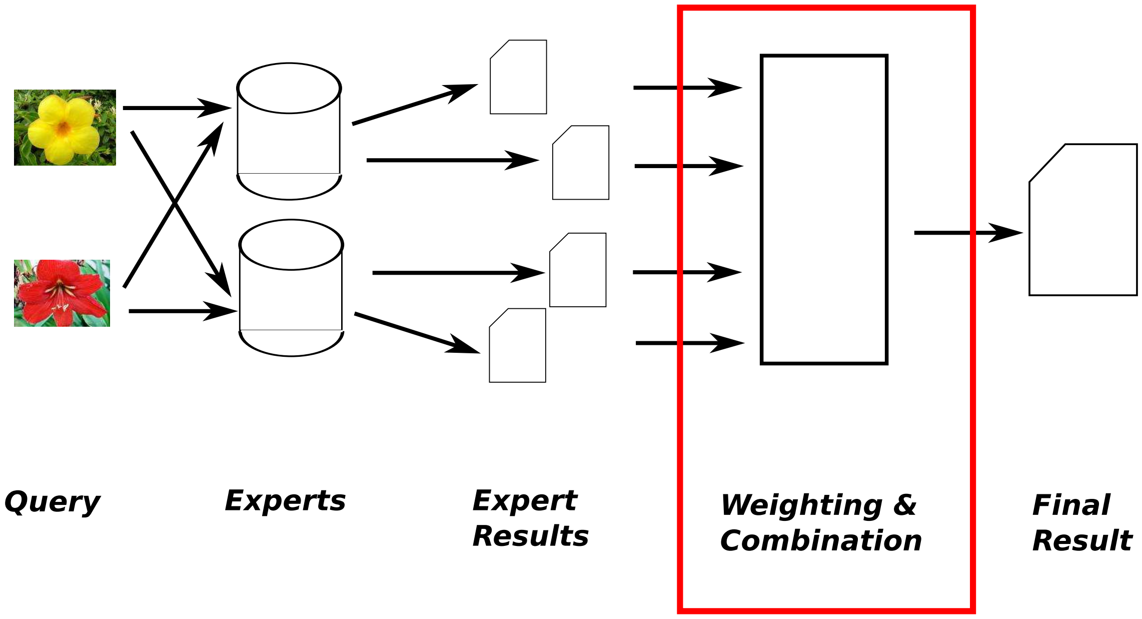


Figure 2.3: CBMIR Investigation Overview

of multiple examples. This will involve multiple visual examples illustrating the information need and a text description of the information need. The paradigm of using visual examples for search is known as 'Query-By-Example' (QBE) (Chang and Hsu, 1992; Jin and French, 2003), where the system is submitted a visual example and is to return results ranked in order of their similarity to the submitted image.

Querying multiple query examples against multiple low-level retrieval experts produces many ranked lists of results which must be combined into a final result for the given information need. The focus of this thesis is given that we are employing multiple noisy retrieval experts, how do we effectively combine these results into a single ranked list, and what factors influence the performance of this combination. Figure 2.3 illustrates the retrieval process utilising low-level retrieval experts and highlights the focus of this work. Central to performance of combining noisy retrieval experts is the implementation of an effective weighting scheme. In this thesis we will be investigating the role of *linear* weighting of retrieval experts for data fusion. Other types of non-linear weightings do exist, however a majority of existing approaches implement linear weighting due to its predictability and performance gains. Part of our work will be examining what is the maximum performance that can be obtained utilising linear weights.

Our CBMIR system will be utilizing text based experts, either derived from ASR, ASR and MT or textual annotations. There are some questions if it is appropriate to address these features as low-level features, as traditionally they are seen as quite a rich source of very descriptive information (Hauptmann and Christel, 2004). We believe that it is appropriate in this case given the wild fluctuations we have seen exhibited by the text experts in our experimental corpora, where often retrieval performance is actually exceeded by individual visual experts. This is because of the noise introduced by both ASR and MT into the quality of the 'signal' of the text expert, and as such we believe it is appropriate that we treat it the same as any other of our sources of noisy data.

2.4 Evaluation

To conduct our experiments into CBMIR utilising low-level retrieval experts we require both multimedia data and a means in which to evaluate our experiments. In this section we will detail the evaluation metrics we will be utilising in this thesis. Firstly however we will detail the resources from which we get our data, the multimedia retrieval benchmarking activities of TRECVID and ImageCLEF.

2.4.1 Evaluation Campaigns

In order to conduct our investigation we require datasets with which to experiment. The datasets which we will utilise come from the TRECVID and ImageCLEF benchmarking activities. Both of these activities follow a setup similar to that established in the Cranfield IR experiments of the 1960's (Cleverdon et al., 1966) where a collection of documents was fixed, along with a test query set and accompanying relevance judgements. This model is considered as a *systems view* of IR, where the needs of the user are represented by the defined topic set and the information satisfying the user's need by the relevance assessments. Typically this form of evaluation is known as the traditional or laboratory IR model (Saracevic, 2007).

TRECVID, formally known as the TREC Video Retrieval Evaluation, is organised by the National Institute of Standards and Technology (NIST) of the United States, and has been running since 2001. The goal of TRECVID is “to encourage research in information retrieval by providing a large test collection, uniform scoring procedures, and a forum for organisations interested in comparing their results” (Smeaton et al., 2006). TRECVID promotes research in content-based video retrieval and compares differing approaches by utilising open, metrics-based evaluation. For video retrieval evaluation to take place, a specification of a common retrieval unit is required. For TRECVID the common retrieval unit is a ‘shot’. A shot is a segment of video that is at least 2 seconds long, and is bookended by cuts, where a cut can be considered as a transition to a visually different segment of video. The evaluation of the relevance of a shot in TRECVID is based only upon the visual information. If only the audio of a shot discusses the search topic but there is no visual evidence, then the shot is considered as non-relevant. For each TRECVID benchmark, NIST provides access to:

- Digital video in MPEG-1.
- MPEG-7 XML detailing the shot boundaries within that video.
- ASR transcripts of the audio from each video.
- If the video was not in English, a Machine Translated (MT) output is provided.
- For evaluations 2003-2006 keyframes extracted from the video. A keyframe is a still image extracted from within a shot boundary as a JPEG image.

The collections we are utilising from TRECVID are the five benchmarks from 2003-2007. Between the years 2003-2006 the data collections of TRECVID for the search activity were comprised of produced broadcast news video. For the years 2003-2004 this video was mono-lingual from English sources, whilst for the years 2005-2006 tri-lingual data was utilised from Arabic, Chinese and English sources. The broadcast news was from commercial sources and included advertisements within the corpora. Broadcast news is characterised by the presence of many anchor person shots which typically bookend news stories within the video. Several research groups leverage data such as this to tune their retrieval performance to exploit this organisation (Hauptmann and Christel, 2004), however in this thesis we employ no domain specific optimisations. The 2007 corpus for TRECVID was a shift away from broadcast news to 'magazine' video. Provided by the Netherlands Institute for Sound and Vision, this corpus was mono-lingual in Dutch and contained documentaries, science shows and news magazine videos. This corpus was quite different to the preceding data collections, with artefacts such as the average shot length being much greater than broadcast news, and the videos themselves being organised differently to broadcast news (Over et al., 2007). An explanation of the sampling strategy for the extraction of keyframes from these corpora is provided in 3.4.1.

ImageCLEFPhoto[1] is a retrieval track activity that occurs within CLEF (Cross Language Evaluation Forum). Similar to TRECVID, CLEF promotes research and development in multilingual information access and is an activity of the Treble-CLEF coordination action under the Seventh Framework Programme of the European Commission. ImageCLEFPhoto specifically is interested in the promotion and evaluation of multilingual visual information retrieval. ImageCLEFPhoto provides a multilingual annotated photograph corpus and like TRECVID compares differing approaches with open, metrics-based evaluation. The corpus used within ImageCLEFPhoto is a subset of the IAPR TC-12 Benchmark, consisting of a collection of 20,000 still natural photographs, each with an accompanying annotation (Clough et al., 2008). Like TRECVID, evaluation of relevance in ImageCLEF is conditional only on the visual data present in the retrieved image, not the text annotation accompanying the image. ImageCLEFPhoto is a track within the ImageCLEF activity, which explores cross-language image retrieval. As part of ImageCLEF they are numerous other tracks which have different objectives to that of ImageCLEFPhoto,

[1] http://imageclef.org

and as such any reference in this thesis to ImageCLEF implies the ImageCLEFPhoto track.

Both evaluations for their respective test topics provided both a textual description of the information need and visual examples. Therefore both evaluations provided data to allow for multi-modal querying. For ImageCLEFPhoto accompanying the query text were always three still natural photographs illustrating the information need. In the case of TRECVID, accompanying the statement there were at least several visual examples which were either still photographs from an external collection, or segments of video from the development collection for that respective year. Where video was given as part of a query we sampled a representative keyframe from the middle of that segment of video and used that as a visual example for search. We selected the middle keyframe from a segment of video as the example video's provided by TRECVID are of shot duration and typically are visually homogeneous. Details of the actual corpora used in our experiments are given in Section 4.1.4.

2.4.2 Evaluation Measures

Throughout this thesis we will be utilising the common evaluation metrics defined in IR literature. Here we provide a summary of the metrics which will be utilised, complete descriptions of evaluation metrics can be found in Van Rijsbergen (1979); Baeza-Yates and Ribeiro-Neto (1999); Blanken et al. (2007). The objective of any IR system is to return for a given information need relevant documents of use in fulfilling the required need. Therefore when we evaluate retrieval algorithms in a laboratory experiment, we are primarily concerned with the degree to which we ranked relevant documents above non-relevant documents, and what relevant documents were excluded from a ranking. The key concepts for evaluation after relevance are *precision*, *recall* and *fallout*. Given a set of returned documents, precision is the degree to which those documents returned were considered relevant

$$Precision = \frac{|relevant \cap retrieved|}{|retrieved|} \tag{2.1}$$

Similarly, recall determines what amount of the total set of relevant documents were retrieved in the result set.

$$Recall = \frac{|relevant \cap retrieved|}{|relevant|} \tag{2.2}$$

Finally, fallout can be considered as the converse of recall, where it determines the degree to which non-relevant documents were returned from the complete set of non-relevant documents.

$$Fallout = \frac{|non - relevant \cap retrieved|}{|non - relevant|} \tag{2.3}$$

Each of these measures can be considered as a set measurement, that is they do not regard the ordering of the documents as presented. Clearly if two systems return the same proportion of relevant documents, but one systems ranks these first whilst the other system ranks these last, then we would like an evaluation metric to reflect the difference between the two, which neither precision or recall can achieve. Average Precision (AP) performs this function, it is designed with a bias towards retrieval runs which rank relevant documents higher in a ranked list, which we formally define in 2.4.

$$AveragePrecision = \frac{\sum_{n=1}^{N} Precision(n) \cdot Relevance(n)}{|Relevant|} \tag{2.4}$$

In AP the function $Precision(n)$ is the precision value at n documents and $Relevance(n)$ is a binary function which indicates if document n is relevant, variable N is the size of the ranked list and $|Relevant|$ is the *total* number of relevant documents in the *collection* under consideration. AP is the most common metric used in the evaluation of IR systems, however it provides a per *topic* score. Mean Average Precision (MAP) is a single scored metric which provides an indication of performance over an entire retrieval run, where a retrieval run consists of the results of multiple topics. MAP is simply the mean of all the AP scores in a retrieval run. For system comparisons, MAP is considered as the most common metric in use in IR research. For our evaluation we also employ statistical testing, coverage of which is provided in section 4.1.1.

MAP is not without criticisms however, notably as it relies upon the mean of the AP scores it is sensitive to outliers, where if there are a handful of queries which achieve exceptional performance, then performance on these topics will dominate the MAP values. One potential solution to this we believe is the application of the standard score, also known as Z-Scores, to the AP values. Z-Scores determine for a given series of observations, how far each individual observation deviates from the mean observation in terms of standard deviations. By calculating the Z-Score of AP (ZAP) and then averaging these values (MZAP), we get an indication for any given retrieval run how far it may deviate from the average retrieval run. The benefit of this proposed metric is that it would discount the potential skewing that MAP can introduce when there are outlier AP measurements in an evaluation. This metric however is only useful when comparing large numbers of retrieval runs, for example the overall evaluation of a TRECVID benchmarking activity. For our given experiments here it is of less use as we do not have a very large set of retrieval runs to compare against as would be the case in a benchmarking activity.

As such for the majority of our experiments in this thesis we will be utilising the metrics of AP and MAP for our comparisons and evaluation. Whilst there may be some concern about the use of these metrics, we believe we alleviate these concerns somewhat by employing multiple experimental corpora (see Section 4.1.4). We will be utilising six corpora for our experiments which range in size and types of data. Therefore if we observe patterns in our experiments across all of the corpora utilised we can have a degree of certainty in the robustness in those observations as being corpora independent.

2.5 Conclusion

In this chapter we have presented a high level overview of the various aspects of CBMIR. We have highlighted that multimedia data is in general a relatively noisy source of information for the purposes of information retrieval. This is due to the effects of both the *sensory gap* and the *semantic gap*. The processing of this data can involve the use of either unsupervised low-level features or supervised high-level features (such as trained concept detectors). We have presented our motivation for restricting our experiments to the use of low-level features, or low-level retrieval experts for experiments in weighted data fusion, notably because we believe that whilst there is significant research being conducted on the construction and application of concept detectors, we believe that low-level features can have substantial impacts upon performance if correctly exploited.

Because each of our low-level information sources is relatively noisy, to obtain good retrieval performance we are required to combine these multiple noisy signals into a coherent response to an information need. To achieve this, we require the use of weighted data fusion so that if a particular source of evidence is performing better than others it can be appropriately weighted in order to improve retrieval performance. The open questions therefore are what factors influence the performance of data fusion, what properties does an ideal weighting scheme for data fusion have, and do existing methods of generating weights for data fusion exploit these properties or can better approaches be developed? In the following chapter we will review the history of data fusion and examine current approaches for weighted data fusion.

Chapter 3

Factors Impacting on Multimedia Retrieval Performance

One of the objectives of this thesis is to conduct an in-depth investigation into weighted data fusion for Content-Based Multimedia Information Retrieval (CBMIR) and to propose, develop, evaluate and access a scheme for the generation of weights to be used for weighted data fusion. In order to reach this objective we first need to study weighted data fusion itself within the context of a CBMIR application. This is necessary as it will allow us to objectively assess what an ideal weighted data fusion scheme should look like and therefore allow us to implement an approach which attempts to match this.

Our investigation into weighted data fusion consists of two key components, the identification and description of what factors impact upon data fusion, and the systematic testing of these factors to measure the impact they have upon data fusion performance. This chapter is concerned with the first of these steps, the identification of these factors which may impact upon performance, whilst the following chapter will present our experimentation and measurement of these factors.

The CBMIR system which we have constructed here is similar to many other CBMIR systems seen in the literature, in that it leverages multiple retrieval experts, each of which can be considered as a relatively *noisy* retrieval expert. The field of

CBMIR itself and indeed successful approaches to it are characterised by the effective combination of these noisy sources of information, such that a response is produced which is of greater performance than that of its constituent parts (Smeaton et al., 2006). Within this context we can see that effective weighted data fusion is crucial to obtaining good levels of performance in CBMIR tasks.

The fundamental action that our CBMIR systems conduct is that of *data fusion*. Data fusion is described by Belkin et al. (1995) as “the combination of evidence from differing systems” with the aim of maximizing retrieval performance. This is as distinct from the *Collection Fusion* problem, which Voorhees *et al.* defines as the combination of “retrieval runs on separate, autonomous document collections that must be merged to produce a single, effective result” (Voorhees et al., 1995).

One of the key features of our CBMIR system, is that it is a *late fusion* system. Within the domain of multimedia search there are two approaches for the combination of data known as early fusion or late fusion (Snoek et al., 2005), these are conceptually illustrated in Figure 4.12. Early fusion is essentially the combination of data prior to indexing, meaning that data is first somehow aggregated, and then a ranking model is placed over this aggregated data. There are multiple examples of this type of system, such as combined document representations, classification tasks or learning to rank applications (see Chapter 5). Therefore the combination of evidence is typically handled in an ‘indexing’ phase. Late fusion instead assumes each source of data has associated with it some form of a ranking function, each of which can be independently queried. Once each source has been queried, the outputs of each of these queries can be aggregated together to form a final response to the initial query. Examples of late fusion include metasearch, and a majority of the text information retrieval experiments of combining ranked lists from different retrieval systems (see Chapter 5). Our choice of employing late fusion for our data fusion experiments means we do not need to explicitly model any particular source of data, as that is the function of the retrieval expert associated with each source of data. This allows us to add or remove retrieval experts at any stage in the retrieval

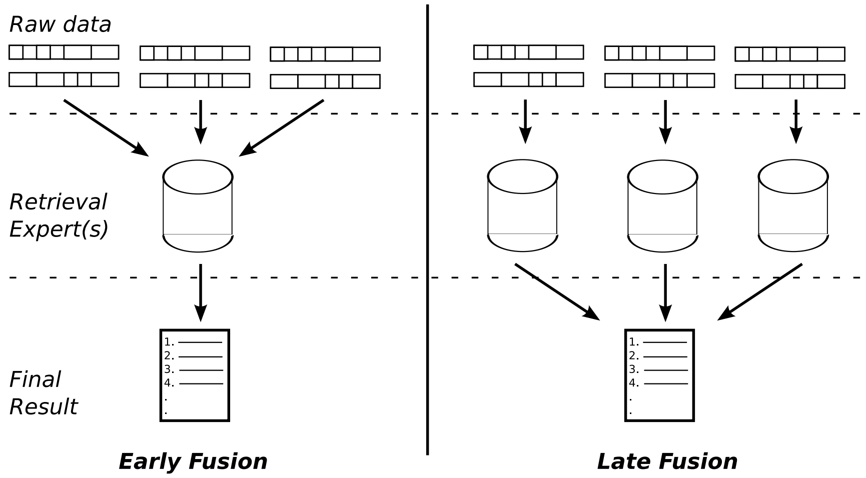


Figure 3.1: Early vs. Late Fusion

process, allowing for a relatively free environment in which to conduct our data fusion experiments. As such this is a generic approach which therefore has wide areas of application. If building an operational retrieval system clearly it would be beneficial to tailor such a system to the retrieval experts employed, however in this study, by keeping a generic approach we seek to demonstrate the impact of ideal late fusion frameworks.

The choice not to study early fusion primarily is that to utilize early fusion, a ranking metric for that combined feature space needs to be developed and tested, typically in other work (Snoek et al., 2005) this would be within a machine learning framework. By studying only late fusion we can treat each source of evidence as independent, as each will have its own data representations and ranking functions, allowing a degree of generality to our methodology as these sources can be added or removed without any re-engineering or other complications arising for our setup. Furthermore this should make our observations more generalized as we are not dependent on the behaviour of any one particular type of ranking function.

As we are using a late fusion framework, our CBMIR retrieval workflow can be described as follows. First we begin with a multi-part multi-modality query, such as two pictures of a flower and the text ‘flower’. Second this query is then sent to each of our retrieval experts, being processed by whichever of the experts are capable of handling parts of the query. Once this is complete we have for each query component from each relevant expert a ranked result list, which we term 'raw' results. Third, as each of these lists needs to be combined somehow we need to perform some form of normalisation that will allow results from each of these ranked lists to be combined. This normalisation may be based on the scores in the ranked lists or on the rank positions. Fourth, we conduct some form of weighting of the ranked lists to be combined, giving greater importance to those results we think are more likely to perform better. Fifth and finally, once the results have been weighted we then combine them, using one of a variety of combination operators or combination levels available to us. Each of these distinct steps in the retrieval process has several different ways in which that step can be achieved. This chapter is concerned with identifying each of these different factors, such that they can be defined and tested in the following chapter. Our retrieval workflow is illustrated in Figure 3.2, and forms a guide for the layout of this chapter.

3.1 Terminology

For reasons of clarity we will now formally define the terms used throughout the remainder of this section. We will assume that a CBMIR system will be combining multiple retrieval experts together for queries which contain multiple components. For these definitions we will be using set and matrix notations, however our use of matrix notation is slightly unorthodox, as matrices typically contain numbers. In our notation, the matrices we define will contain ranked lists of documents. The use of matrix notation is for conceptual reasons to assist the reader in comprehending the number of variables at work.

3.1.1 Retrieval Expert

A Retrieval Expert is treated as a black box, it is a service which can process a query and return a ranked set of documents on a given collection. Formally our

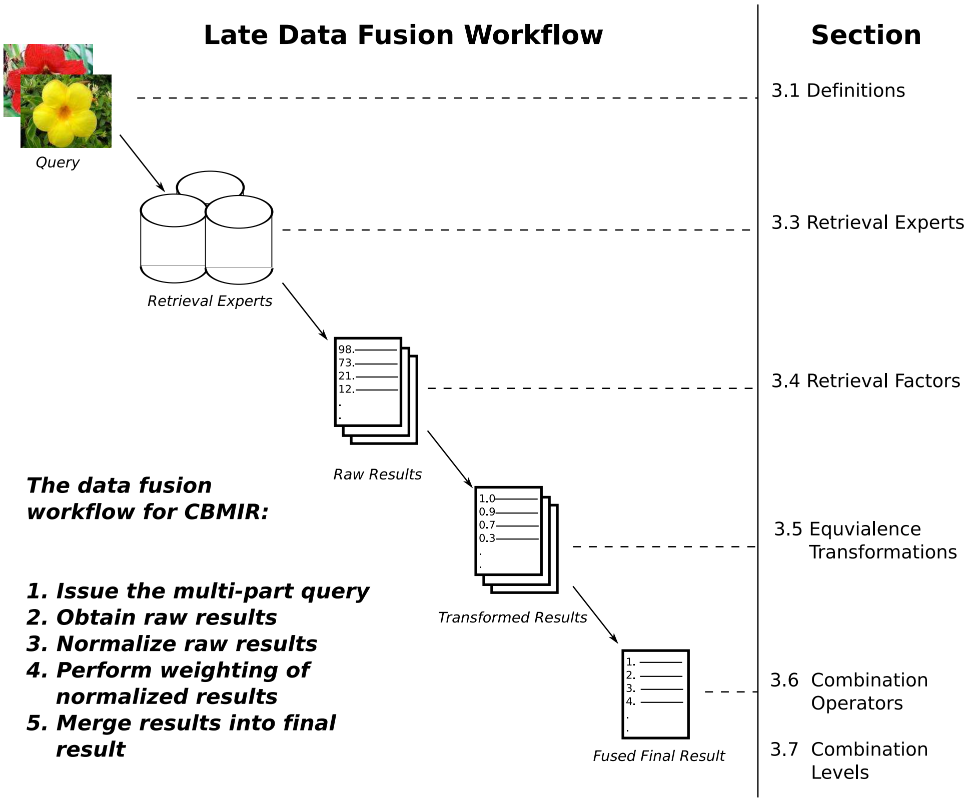


Figure 3.2: Data fusion workflow, from the issuing of a multi-modal query, through to the computation of a final result.

system will have Retrieval Experts $E = \{expert_1 \; ... \; expert_i\}$ where $1 \leqslant i \leqslant |E|$.

3.1.2 Query

A Query within our context is a multi-modal request which describes an information need and is to be processed by the CBMIR system. A query may be comprised of text, multiple visual examples, etc. Within this context, each component of a query is treated as a separate query to be processed, e.g. a text component would go to the text expert, each individual visual component would go to visual experts. Therefore a Query Q is comprised of $\{query_1 \; ... \; query_j\}$ where $1 \leqslant j \leqslant |Q|$. As an example, if a user is searching for pictures of boats, the query may be the text 'boats' and two images of boats, therefore in this case the individual components of the query are $\{`boats', image_a, image_b\}$ otherwise referenced as $\{query_1, query_2, query_3\}$, each of which may be individually processed by the retrieval experts available.

3.1.3 Documents

Documents in this context are the semantic unit of information that is indexed and retrieved. Typically the term refers to entire text documents or webpages for ranking. In the case of MIR, the unit of retrieval can be a video, a 'shot' (i.e. a segment of video which is visually consistent), audio recording, etc. The use of the term 'documents' will refer to any retrieval unit that can be handled by a CBMIR system.

3.1.4 Result Set

A result set is the product of a unique pair of Retrieval Expert and Query $\langle expert_i, query_j \rangle$, which produces an ordered set of documents R such that $R = \{document_1 \; ... \; document_m\}$ where $1 \leqslant m \leqslant |R|$. Every unique pair $\langle expert_i, query_j \rangle$ produces a Result Set, which collectively form the matrix $\mathbf{RS} = [rs_{i,j}] \; i = 1 \; ... \; |E|, j = 1 \; ... \; j = |Q|$. The row index i represents experts, while column index j represents query components. Therefore, $rs_{i,j}$ represents the result set R generated by expert i and query component j.

Every $document_m$ in R will have associated with it a triple $\langle name, rank, score\rangle$ where *name* is a unique identifier for the document, *rank* is in the set $\mathbb{N}$ and *score* is in $\mathbb{R}$. Every value of *rank* will be unique in the set and it is desirable that every *score* value is also unique, however depending on the ranking function this may not always hold. As the set R is ordered, the values of both *rank* and *score* will change monotonically as one iterates through the set. In order to compute a final response to a query, a set of coefficients is typically required so that we can give greater weight to those sets R which are likely to enhance retrieval performance.

3.1.5 Retrieval Coefficients

We define a matrix of weights which are used to alter the impact of different $rs_{i,j}$ when they are combined into a single result. This matrix is **RC**, where $\mathbf{RC} = [rc_{i,j}]i = 1 \ldots |E|, j = 1 \ldots j = |Q|$. Individual coefficients have the properties $rc_{i,j} \in \mathbb{R}^+$ and $\sum rc_{i,j} = 1$. Every result set in matrix **RS** will have a corresponding entry in **RC**. For instance if no weighting was desired, all entries in **RC** would be set to the same value, thus providing a uniform weighting.

The final result set therefore, is some combination of the result sets rs_{ij} generated by pairs $\langle expert_i, query_j\rangle$ from sets E and Q, and application of the retrieval coefficients **RC**. Taking our previous example of finding images of boats with an CBMIR system which has available 3 retrieval experts (one text expert, two visual), we have potentially 5 result sets in which to combine into a single result set (one text result and four visual results), and up to 5 weights that can be applied.

3.1.6 Example CBMIR System

To help illustrate the various approaches that different weighting schemes employ and how they impact upon retrieval, we will refer back to this section to help il-

Example System and Multi-Example Query			
E	*Expert Set*	*Q*	*Query Set*
$expert_1$	Text Expert	$query_1$	"Flowers"
$expert_2$	Colour Expert	$query_2$	
$expert_3$	Edge Expert	$query_3$	
$expert_4$	Texture Expert		
$\lvert E \rvert$	4	$\lvert Q \rvert$	3
Result Set matrix **RS**			
$rs_{1,1}$	"Flower" ↦ Text Expert	$rs_{3,2}$	↦ Edge Expert
$rs_{2,2}$	↦ Colour Expert	$rs_{3,3}$	↦ Edge Expert
$rs_{2,3}$	↦ Colour Expert	$rs_{4,2}$	↦ Texture Expert
		$rs_{4,3}$	↦ Texture Expert
Non-zero entries in matrix **RS** : 7			
Retrieval Coefficients matrix: **RC** rows *i* experts, columns *j* query components			
$\begin{pmatrix} rc_{1,1} & 0 & 0 \\ 0 & rc_{2,2} & rc_{2,3} \\ 0 & rc_{3,2} & rc_{3,3} \\ 0 & rc_{4,2} & rc_{4,3} \end{pmatrix}$			
Non-zero entries in matrix **RC** : 7			

Table 3.1: Example system and query, where there are 4 experts in the system and the query has 3 components.

lustrate how various approaches may work. Defined in this section is an example CBMIR system and a multi-example query to it, as shown in Table 3.1.

In this example system, there are four retrieval experts (E) available within the system (where each of these experts is considered a black box with its own index and ranking function). We have also defined a query (Q) that is issued to the CBMIR system, which consists of the text "flowers", an image of a red flower and an image of a yellow flower. The system presented with this query, generates seven result sets (R), one for the text query, then six more from the two query images against the three visual experts. Technically there are more instances R than listed here, such as querying a visual expert with the text query, however this will produce a null set of results which for reasons of clarity we do not show here.

3.1.6.1 Terminology Summary

To summarise, we have defined within our CBMIR system, retrieval experts E, multi-part queries Q, individual result sets R and the matrix which contains all result sets $\mathbf{RS}$. To weight each ranked list contained in the matrix $\mathbf{RS}$ we have the retrieval coefficients matrix $\mathbf{RC}$ which is used to weight each ranked lists so that they can be combined into our final response to a query.

$$E = \{expert_1 \ ... \ expert_i\}$$

$$Q = \{query_1 \ ... \ query_j\}$$

$$R = \{document_1 \ ... \ document_m\}$$

$$document_m \mapsto (name, rank, score)$$

$$\mathbf{RS} = [rs_{i,j}]_{|E| \times |Q|}$$

$$\mathbf{RC} = [rc_{i,j}]_{|E| \times |Q|}$$

3.2 Previous Studies

There have been previous studies into the factors which impact upon data fusion in Information Retrieval. A review of data fusion literature and current techniques for generating weights for data fusion is presented in Chapter 5. Here we briefly summarise work which has explicitly looked at what factors impact on data fusion and retrieval performance.

Early work into the investigation of data fusion and information retrieval was conducted by Lee (1997) who combined text experts together to achieve a performance gain. His study on what were the driving factors which caused data fusion to work focused on an examination of the relevant documents found in common between ranked lists, and the non-relevant documents in common. Exploring this he defined two measures, $R_{Overlap}$ and $NR_{Overlap}$, which measures the degree to which expert results agree on relevant and non-relevant documents. These measures are presented as Equations 3.1 and 3.2.

$$R_{Overlap} = \frac{R \cap S_1 \cap S_2 .... \cap S_n}{(R \cap S_1) \cup (R \cap S_2) \cup ...(R \cap S_n)} \tag{3.1}$$

$$NR_{Overlap} = \frac{NR \cap S_1 \cap S_2 .... \cap S_n}{(NR \cap S_1) \cup (NR \cap S_2) \cup ...(NR \cap S_n)} \tag{3.2}$$

where $S_1 ... S_n$ are ranked lists being combined. $R_{overlap}$ measures the intersection of relevant documents between ranked lists over the set of all relevant documents, whilst $NR_overlap$ does the same for non-relevant documents. Lee's finding was that data fusion appeared to work between good quality text retrieval experts as good quality ranked lists shared similar sets of relevant documents, but dissimilar sets of non-relevant documents. Therefore when combined, the common relevant documents are promoted up the ranked list (Lee, 1997). Beitzel et al. (2004) however examined Lee's hypothesis and draws different conclusions. Performing a systematic approach to identifying individual factors which may impact on data fusion, an approach which we emulate, Beitzel found that when combining different *ranking models* where stemming and stopping are held constant, that effective ranking models already highly rank relevant documents and that system performance was degraded as common *non-relevant* documents were promoted up the ranked list. Therefore, whilst the combination of ranked lists from *retrieval systems* improved performance (where systems may have different stemming, stopping etc), combination of high-performing ranking models alone did not produce an improvement. Nevertheless the definition of the two overlap measures provides a good mechanism for inspecting the effect of data fusion. Furthermore in the case of CBMIR we are using *weak* retrieval experts, quite distinct in performance from traditional text ranking models, where weighting must be employed to achieve good performance.

Further examinations of data fusion in text retrieval include work by Montague and Aslam (2001) who proposed and investigated the use of various score normalisation approaches for ranked list combination (see Section 3.5.1). Hawking and Robertson (2003) investigated the effect of the size of the collection on retrieval performance. Robertson et al. (2004) provided an examination of attempts to introduce weighting of specific fields of text documents into the BM25 ranking algorithm. Robertson *et al.* found that to effectively weight field components using BM25, that weighting should be conducted before the ranking model calculations, effectively an instance of early fusion.

Likewise for multimedia there have been several investigations into data fusion and its effects on performance. Yan and Hauptmann (2003) conducts a theoretical investigation into the use of linear weighting and expert combination for multimedia retrieval. McDonald and Smeaton (2005) performs a thorough empirical investigation into the differences of using scores, ranks and probability fusion methods for multimedia retrieval. Urban and Jose (2004) conducts an investigation into various statistical combination strategies for multiple example visual queries. de Sande et al. (2008) compares the use of various colour experts and their performance for the task of object and scene recognition.

There are several key distinguishing factors that differentiate our work from these examples just provided. Firstly, in comparison to many of the text data fusion investigations, we are using very weak retrieval experts, where weak is a reference to typical performance of these experts as measured by MAP. Because of this, not only are we investigating the combination of these experts but also the best distribution of linear weights which must be employed in order to obtain the best performance possible from the inputs provided. Secondly, a majority of the multimedia investigations contain some implicit level of expert aggregation (see Section 3.7), such as aggregating the results from a colour expert together and then combining those results with the result of a text expert. Thirdly, to our knowledge, we are unaware of work which examines data fusion for multimedia retrieval examining all of the variables which we will identify in this chapter. Typically in previous investigations (Yan and Hauptmann, 2003)(Urban and Jose, 2004)(McDonald and Smeaton, 2005), some aspect is fixed and assumed to offer no variation in performance. Furthermore, the novelty of our work is in taking all of the factors we identify in this chapter, and conducting an optimisation process directly on the test data of a collection using these factors. This process, discussed in the next chapter, will allow us to make absolute observations about the impact various factors may have on retrieval performance as we will first find what the optimal weighting combination for that set of factors should be.

The first stage in identifying what factors may impact data fusion performance in CBMIR is the definition of the retrieval experts we will be using for this study.

3.3 Retrieval Experts Used

In this section we will briefly detail the retrieval experts that our CBMIR system will employ throughout our investigation into data fusion and multimedia retrieval. This is not intended to be a comprehensive overview of the workings or capabilities of these retrieval experts, but rather an overview of what is being used. Whilst in previous parts of the thesis we have commented on the broad range of components that can be integrated into a CBMIR system, for our implementation we focus on two distinct classes of retrieval experts, text experts and visual experts. This restriction is more of a practical consideration as a consequence of the feature extraction tools available within out research group. Notable features which could also have been included are motion and audio experts, however these experts would not have been of use on the ImageCLEF corpora. In a similar vein, an operational retrieval system presumably would have some expert knowledge about the collection being indexed. In this work we make no accommodation for collection specific tweaks or enhancements, again employing a generic framework. However an operational system would be foolish not to undertake some tailoring towards the collection being indexed so as to enhance retrieval performance.

3.3.1 Text Experts

Text retrieval experts have traditionally been one of the best performing retrieval experts in multimedia retrieval within the benchmarks we are exploring (Hauptmann and Christel, 2004), and for that reason we employ a text retrieval expert in our work. A text retrieval expert, as the name implies, works off text documents on which to index and retrieve. Depending on the benchmarking collection used, what comprises those text documents differs. The ImageCLEFPhoto 2007 collection (Clough et al., 2008) provided multi-lingual image annotations which accompanied every image. These annotations could be considered as being of very high quality, they are free of any noise which may distort the documents, and so offers good performance when retrieved.

For digital video data the picture is somewhat different. The text available for indexing from video is typically via Automatic Speech Recognition (ASR) (Blanken et al., 2007; Huijbregts et al., 2007). This process uses statistical models to take an audio signal from digital video and generate a text transcript of any speech from within that video. Whilst the accuracy of ASR techniques are continuously improving, they are inherently more noisy than a manually created transcription of the audio. The TRECVID 2003, 2004 and part of the 2005 and 2006 collections have available straight ASR transcripts from English speech. However the TRECVID 2005, 2006 and all of the 2007 collections have video in which other languages are also used. This means that the ASR being detected is in a language other than English, and first requires translation into English for it to be of use for our retrieval system. As our search topics and would be users speak English, being able to process English queries is a requirement of this system. The task of translation is given over to Machine Translation (MT) techniques which automatically translate the text from its original language into English. However, this process introduces a second layer of noise to the text expert, as the raw data is first extracted from a noisy process (ASR), then is transformed through an additional noisy process (MT) to be ready for indexing. We will see at the end of this section how the performance of text varies throughout the various benchmarks we investigate.

The text retrieval expert which we utilised for the TRECVID collections is a vector space model, implemented within the Terrier search engine (Ounis et al., 2007). All text documents are stemmed and stopped. For the ImageCLEF collection, our text expert uses English query to English document elements of the text corpus. The ImageCLEF text retrieval results we used were implemented as part of a previous collaboration we had with the University of Tampere, Finland, and utilised a language modelling approach, implemented by the Lemur toolkit (Metzler et al., 2006; Järvelin et al., 2007). The utilization of the vector space model for our work was primarily that the text documents we are using are typically very short, and as such the vector space model offered good performance with minimal parameter tuning.

A complete review of the various text retrieval models which could be utilised in retrieval is beyond the scope of this work, however many texts exists, such as Van Rijsbergen (1979); Salton (1989); Witten et al. (1999) which provide extensive coverage of the area of text retrieval.

3.3.2 Visual Experts

Visual experts are experts which given a visual source of data, typically an image, seek to describe that image from some particular viewpoint, such as the colours distributed within the image or whatever textures may be present. Once an image has been described by a particular expert it is then available for retrieval. The visual experts which we utilise in this thesis can be referred to as using 'low-level' features. That is, they describe the visual data in terms of some transformation or statistical inference about the data, but they do not make any semantic inferences about the visual data. The extraction of 'low-level' features from a visual corpus is unsupervised, and can be thought of as an analogue to term extraction techniques from text information retrieval. For instance, ‘low-level’ visual features may tell us that an image is predominately blue, with a bright yellow circle in the top left corner. What a ‘low-level’ feature will not tell us is what this image may represent, which is typically the task of semantic concept detectors (see Chapter 2).

The majority of the visual experts we employ in this work are derived from the MPEG7 XM (eXperimentation Model - see 2.2.1), a reference implementation of features described in the MPEG7 standard (MPEG-7, 2001). The MPEG7 standard is a multimedia content description interface whose objective was to standardise multimedia content descriptions to improve interoperability between multimedia systems. Visual features within MPEG7 are referred to as ‘descriptors’, and these are typically compact descriptors, that is they are not overly verbose in describing the visual content of an image. The implementation of the MPEG7 visual descriptors which we use was created within our research group as part of the aceMedia project and based upon the MPEG7 XM (O’Connor et al., 2005). The similarity metrics we use to facilitate querying and ranking of visual features are those defined within the MPEG7 XM and are typically variations on geometric distances such as Euclidean distance. We will now briefly describe the six visual features which we will utilise in our experimentation. The visual experts we define here will either have *global* or *spatial* properties. Global properties can be thought of as giving details of the overall image as a whole, whilst spatial properties would segment an image into smaller regions within an image and describe each of those (e.g. overlaying a 3×3 grid over an image and examining each square).

Further details on the MPEG7 features can be found in (O’Connor et al., 2005; Manjunath et al., 2002) whilst general details on low-level visual features can be found in Blanken et al. (2007). Within our work we utilise two broad classes of visual experts, colour experts and texture experts. We selected these six experts so as to give a degree of variability within the experts with regards to their performance. There is no standard number of experts employed within the research literature, many systems may make use of only one feature, whilst others which have supercomputing resources at their disposal will make use of several, particularly for classification tasks. Our use of six visual features we would consider to be on the higher end of the scale with regards to the number of experts typically used.

3.3.2.1 Colour Visual Experts

Colour visual experts are among the most popular of the visual experts, they provide an intuitive method of querying a database and can offer good performance depending on the task and the collection (Blanken et al., 2007; Datta et al., 2008). However there are many different ways in which colour can be represented, each having advantages or disadvantages given the particular retrieval task. Therefore we employ a range of colour features which capture different aspects of colour, such as global colour averages, which colours occur in which regions or even altering the colour space that is used to represent the colour. A discussion on colour and how we perceive it with its impact on various computing applications can be found in Humphreys and Bruce (1989). We will now discuss the visual experts which we utilised.

Scalable Colour Descriptor is derived from a colour histogram defined in the Hue-Saturation-Value (HSV) colour space. It uses a Haar transform coefficient encoding, allowing a scalable representation, and is constrained to 256 bins. This is a global visual expert and can be thought of as a more compact standard colour histogram.

Colour Structure Descriptor is also based on colour histograms, but it aims at identifying localised colour distributions using a small structuring window. In other words, it represents an image by both the colour distribution (similar to a colour histogram) and the local spatial structure of the colour. Therefore it aims to capture elements both of a global and spatial nature.

Colour Layout Descriptor is a very compact descriptor which captures the spatial layout of the representative colours on a grid superimposed on an image. It is designed to efficiently represent spatial distribution of colours. The image is divided into an 8×8 grid, and the average colour in each region is recorded.

Colour Moments Descriptor This descriptor provides a means of describing the colour of an image which is alternative to the Colour Layout descriptor. An image is divided into 4x4 subimages and for each subimage the mean and the variance on each is computed. Whilst Colour Layout utilises the YCbCr colour space, the Colour Moments descriptor utilises the LUV colour space.

3.3.2.2 Texture Visual Experts

Colour, whilst useful in image retrieval, is not the only source of visual experts which we will utilise. The second class of visual expert we will employ are known as texture experts. Texture experts are useful because they extract patterns from the visual data, for instance an aerial photograph of a full car park would demonstrate a strong pattern which could be searched against to find other similar images regardless of the colours used in those images.

Edge Histogram Descriptor This descriptor represents the spatial distribution of edges in an image. The image is divided into 4 x 4 subimages and the local-edge distribution for each subimage is represented by a histogram. To generate the histogram, edges are categorised into five types: vertical, horizontal, 45 degrees diagonal, 135 degrees diagonal and non directional.

Homogeneous Texture Descriptor Provides a quantitative representation using 62 numbers, consisting of the mean energy and the energy deviation from a set of 30 Gabor frequency channels.

3.3.3 Expert Performance

For illustrative purposes we conduct a simple experiment here where we measure the performance of each individual expert against each of our test corpora. The

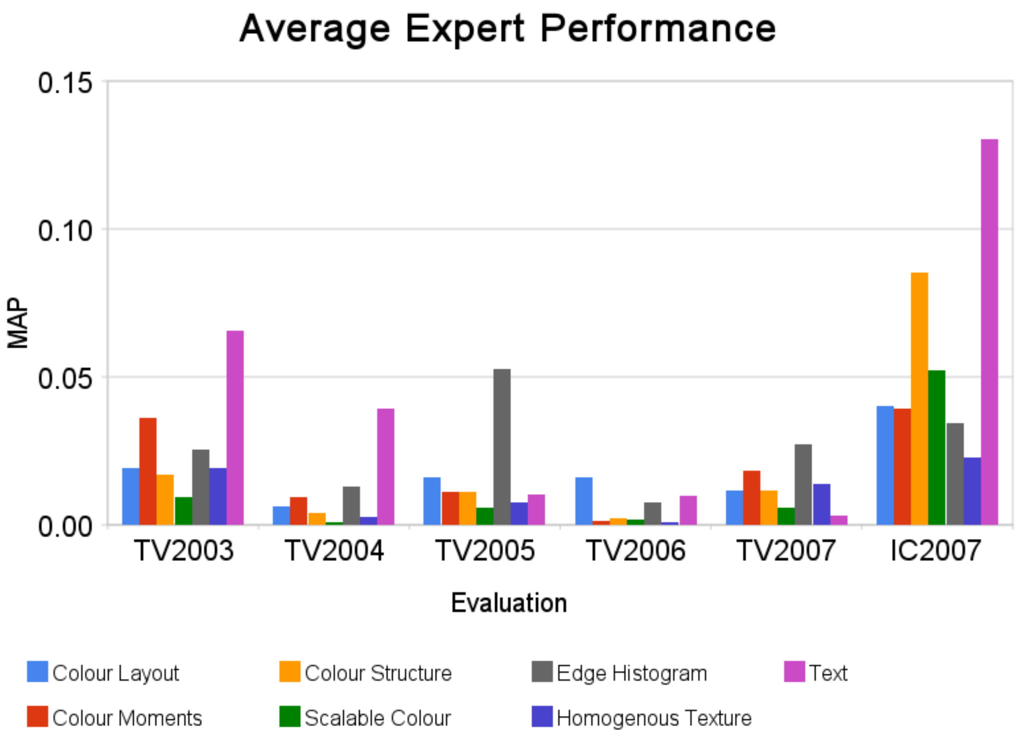


Figure 3.3: Average Expert Performance, Chapter 4 will highlight the performance gains of correctly combining these features.

specifics of this test are that for each expert if there are multiple query-components (e.g. the query contains multiple visual examples), then the results of these are uniformly weighted and combined to provide an average indication of that expert's performance. The results of this experiment are presented in Figure 3.3, with the X-axis representing the different test corpora which we will be using, and the Y-axis representing Mean Average Precision (MAP). These results were generated by utilizing MinMax normalization and CombSUM for combining multiple query-components within the one expert (i.e. as our queries are typically made up of three or more images, the result for a single expert is the aggregation of the results of those three queries for the one expert).

From this graph we can draw several conclusions. The first conclusion is that individual retrieval experts perform very badly in terms of MAP across the majority of evaluation corpora. This is demonstrated by the majority of the lines for each expert residing in the range 0.001 - 0.03 of MAP. The second conclusion is the variation in the text expert, correlating with the corpora. We see better performance for TRECVID 2003 and 2004, however for the multi-lingual and Dutch collections of TRECVID 2005, 2006 and 2007 we see a large deterioration in performance. Conversely with the ImageCLEF 2007 corpus, the only corpus with noise free text, we see that the baseline text expert performs well. Coupled with this is our final observation on the corpora themselves, where we note that there is significant performance shifts across corpora with different experts, re-enforcing the selection of multiple corpora on which to test so as to hopefully gain more generalised knowledge of the behaviour of data fusion across a wide range of retrieval environments. We would note that this section is to illustrate the relatively poor performance of low-level features when used for retrieval in isolation. Chapter 4 will demonstrate how these features when combined achieve excellent retrieval performance.

3.4 Retrieval Factors

Key factors that can influence the outcome of data fusion in CBMIR, involve the granularity at which we sample from video keyframes for indexing, and secondly, for each individual retrieval expert, how many results we request from each expert. We will refer to these two factors as the *sample rate* and the *read depth.* These factors come into play when a request is made of an expert to provide results for a query.

3.4.1 Sample Rate

As we previously identified in Chapter 2, some of the multimedia data collections we will be using came from the TRECVID benchmarking activity (Smeaton et al., 2006). TRECVID is primarily concerned with the advancement of techniques for content-based retrieval of digital video. Digital video as previously discussed is a temporal medium, where frames (images) are presented at a rate of a least 25 frames per second to give the perception of a continuous view (Blanken et al., 2007). For the visual experts we use in our CBMIR system, we must sample frames from the video which we can then index so we can have discrete retrieval objects. Therefore the *sample rate* which we utilise may have an impact upon the performance of different retrieval experts.

Again as previously discussed (Chapter 2), the unit of retrieval when working with TRECVID data is the 'shot', which is a segment of video which is visually similar and has a duration of at least 2 seconds in length (Smeaton et al., 2003). However, within 'shots' there may be 'sub-shots', which are segments of video that are visually dissimilar to the 'shot' in which they are contained. They are less than 2 seconds in duration however, so are aggregated into a shot. For TRECVID 2003-2006, NIST provided keyframes as part of the data set to be used for each year's evaluation. There were two groups of keyframes, RKF and NRKF. RKF keyframes are Representative KeyFrames (RKF) and are a single keyframe extracted from the temporal centre of a shot. Non-Representative KeyFrames (NRKF) represent images from 'sub-shots', where this frame was taken from the temporal centre of that 'sub-shot'. In TRECVID 2007, NIST provided no official keyframe set as part of the official data set, leaving it up to individual research groups to select the sampling strategy they wished to implement (Over et al., 2007). As part of our research group's participation in TRECVID that year, we implemented an aggressive sampling strategy, extracting frames at a rate of approximately 1 frame from every 30 frames of video. Specifically we sampled every second I-Frame, where the I-Frame, known as the Intra-Frame and defined in the MPEG standard, is a frame which can be decoded independent of any other frame (Smeaton, 2004). The resulting frames extracted are referred to as K-Frames (Wilkins and et al., 2007). Therefore within our video corpora, we have three levels of sampling strategy available to us for indexing video, which are:

- **RKF** One keyframe per shot.

- **NRKF** At least one keyframe per shot.

- **K-Frame** Regular temporal sampling of frames, many frames per shot.

For our experiments into factors influencing data fusion we will examine these different levels of keyframe sampling. We will investigate the levels RKF and NRKF in TRECVID collections 2003-2006 and RKF and K-Frame in the TRECVID 2007 collection. For clarification, the retrieval unit is the 'shot', so when we retrieve results from a visual expert we aggregate results to the level of the shot. Using the RKF sampling strategy this is straightforward as there is only ever one keyframe per shot, therefore the result of an RKF keyframe is the result of the shot. For the NRKF and K-Frame sampling strategies we implement MAX behaviour, where if for a given query we have multiple results from a single shot, as the retrieval score for that shot we take the value of the highest ranked keyframe. In unpublished experimental work we also attempted using the average and MIN behaviour for aggregation, and found that MAX behaviour provided the best results with regards to retrieval.

3.4.2 Read Depth

We define *read depth* as the number of results we request from a retrieval expert for any given query. In text retrieval there is known to be an inverse relationship between recall and precision, such that as recall increases precision will decrease (Buckland and Gey, 1994). However as we request more results from a given retrieval expert, recall will increase as we are obtaining more documents from the collection. Typically, however, we perform some truncation of the results we request from a given expert, otherwise if we do not and return the entire collection indexed, we would get the benefit of 100% recall but also return a very large amount of non-relevant data (Van Rijsbergen, 1979; Salton, 1989; Buckland and Gey, 1994). Clearly the size of the result sets we request from any given expert will impact upon performance.

Traditionally in benchmarking activities such as TREC, the truncation of result sets is typically the top 1000 results for *ad-hoc* retrieval tasks (Harman, 1993). The task of CBMIR utilising weak retrieval experts is very dependant on data fusion in order to achieve good retrieval performance. A broader question therefore is what is the impact of result set truncation on data fusion within the context of CBMIR, and indeed how does this vary as we either increase or decrease the size of result sets to be combined through data fusion. Therefore an additional set of variables we will examine for their impact on data fusion, is the depth to which we read results from retrieval experts, and how this varies as we add or subtract retrieval experts.

3.5 Equivalence Transformations

One of the fundamental challenges of data fusion for Information Retrieval is the combination of result sets generated from different retrieval experts. As each expert is essentially a separate retrieval system, it will have its own ranking function and as such the result sets that an expert may generate may be quite different in distribution to that of another expert. For instance, a visual expert which uses a dissimilarity metric for ranking would produce ranked lists where the scores are ranked in ascending order, with the lowest score being the best, whilst a probabilistic system would rank documents by score in descending order, with the highest score being the best. What we can assume from retrieval experts is that the lists generated are indeed ranked, and that the change in scores through a ranked list is monotonic, that is that the documents should be ranked in order of their likeliness of being relevant for a query (Robertson, 1977). There are two broad classes of transformations which we can apply to ranked lists in order to allow them to be easily combined. These are rank based transformations and score based transformations.

3.5.1 Score Normalisation

As previously identified, when a retrieval expert ranks a document, we have access to the document's rank and score assigned by the expert. The score of a document is often seen to be a desirable attribute to work with, as it is assumed that the score provides additional information such as the distribution of scores from an expert or the strength of a ranking decision (McDonald and Smeaton, 2005; Yan and Hauptmann, 2003; Renda and Straccia, 2003). The difficulty however is in combining multiple sets of scores together, as the scores may have different ranges or different sortings (i.e. ascending or descending). We require therefore methods to normalise the scores, such that they can be combined.

3.5.1.1 Z-Score

A second normalisation strategy comes from general statistics and is known as the standard score or the Z-Score (McClave and Sincich, 2006). Whilst this normalisation strategy has seen some use (Renda and Straccia, 2003; Montague and Aslam, 2001) its general use in data fusion appears less widespread than MinMax. Z-Scores are also known as shift and scaling normalisation, where given a list of scores to be normalised, we first shift the mean of the scores to 0, then scale the scores such that the standard deviation of the scores becomes 1. This is shown in Equation 3.3.

$$Norm_{score(x)} = \frac{score_x - \mu}{\sigma} \tag{3.3}$$

where μ is the average score of the ranked list, σ is the list's standard deviation and $score_x$ is the value to be scaled. Like MinMax, Z-Score normalisation is a linear transformation of the scores as it preserves the shape of the distribution of those scores, however as identified by Montague and Aslam (2001), this normalisation process has several key differences to MinMax normalisation. The major difference is in the scale transformations where Z-Score transformations are scale invariant. For simplicity, if we assume that the scores of a ranked list are normally distributed, then we could expect the range of the Z-Score values to span $[-3 : 3]$. However there is no fixed maximum or minimum value as there is in MinMax normalisation (1 and 0 respectively).

Therefore we have two score normalisation approaches, both of which are linear transformations of the scores, but have variability in the scaling of the range of the values that scores can take. In MinMax normalisation, as the top ranked score becomes 1, it means that when combining against multiple experts, each expert is treated uniformly, in that each expert's top ranked score has equivalent values. Alternatively, Z-Score normalised scores preserve the scale of the initial ranking. If a ranked list has very similar scores with no outliers, we would expect the scores to be clustered within a very tight range, whilst if a ranked list is populated with outliers, the range of scores could be very broad. This would impact when combining multiple ranked lists together, as the top ranked score of each ranked list is no longer equivalent.

Another class of transformation which is guaranteed to provide equivalence for ranked lists when combining them, is to perform transformations on the ranks of the documents.

3.5.1.2 MinMax Normalisation

The first normalisation strategy we will examine is known as *MinMax* normalisation and has seen extensive use in the field of data fusion beginning with Fox and Shaw (1994). MinMax is defined in Equation 3.4.

$$Norm_{score(x)} = \frac{Score_x - Score_{min}}{Score_{max} - Score_{min}} \tag{3.4}$$

Assuming that our ranked list of results is sorted, $Score_{min}$ is the value of the last element in the ranked list, $Score_{max}$ is the top ranked element, and $Score_x$ is the current score being normalised. This transformation is a linear transformation which produces a set of scores in the range [1 : 0], where the top score is guaranteed to be 1 and the lowest score is 0. As this transformation is linear it preserves the shape of the distribution of the scores.

3.5.2 Rank Transformations

Rank-based transformations only consider the rank information of a ranked list, and as such the scores of a ranked list are only required in order to generate the ranking of the list. Popular in metasearch applications, where the assumption is often made that the aggregator will only have access to the ranks of documents from a search engine and not its scores, rank transformations can take several forms. Rank transformations have been examined by several authors including Jeong et al. (1999); Dwork et al. (2001); Renda and Straccia (2003); Aslam and Montague (2001). Operationally we can consider rank transformations as the same as score transformations, in that we take in some raw value (in this case the initial rank as opposed to score) and apply some form of transformation which generates a new 'score' which we can be used to combine documents between ranked lists.

3.5.2.1 Borda Count

One of the most well known and utilised rank transformation techniques is Borda Count. Initially developed as a voting method in the 18th century as an implementation of preferential voting, it has seen application in Information Retrieval tasks (Aslam and Montague, 2001). Given a ranked result set of documents rs of length N, the Borda method in its simplest form is that for any given document x, subtract the rank of x from the value of N. Formally this is expressed in Equation 3.5.

$$Norm_{score(x)} = N - rank_x \tag{3.5}$$

For example, given a ranked list of 1000 documents, the top ranked document would have a Borda count of 999, the second ranked document a Borda count of 998 etc. This transformation like the previous score transformations is linear, however problems are encountered when ranked lists are to be combined which are of different lengths. For instance, if we have one ranked list of size 1000, and a second ranked list of 100 documents, then the top ranked document in the first list will be given a score of 999, whilst the second list's top ranked document will only be given a score of 99. Depending on the data fusion application, this may or may not be a desirable quality, a short result list may indicate that few documents were found and therefore are unlikely to be relevant. Conversely it may indicate the presence of a very precise query and the small set of returned results are of a high quality. Like Renda and Straccia (2003) we briefly propose two extensions to the basic Borda method to address the list imbalance problem. Whilst the work of Renda *et al.* extended Borda by assigning non-ranked documents a uniform low score, our approach differs as the documents returned from our experts may be quite different sets as opposed to different text experts which work off the same symbols (i.e. words).

The first extension is given that multiple lists are to be transformed, we first determine what is the size of the largest ranked list to be normalised, and use this value instead of N from which to subtract the current rank. We will refer to this approach as BordaMAX, and it is formally given in Equation 3.6:

$$Norm_{score(x)} = max(|rs|) - rank_x \tag{3.6}$$

where the function $max(|rs|)$ returns the size of the largest ranked list being normalised. Taking our previous example of two ranked lists of size 1000 and 100, with this approach the top ranked documents of both lists are assigned the score 999. This in effect creates the opposite problem to that which we discussed with Borda count, as short ranked lists relatively speaking are having greater impact than longer lists. Again however this is a decision up to the system implementer to decide which approach is more suited for their retrieval task.

Our second approach takes a middle ground between Borda count and BordaMAX, which we term rankMM. The rankMM method is the traditional Borda count method applied, followed by a normalisation of the resulting Borda count values by applying MinMax normalisation. This final extension has the property that all top ranked documents from the ranked lists to be combined will all have the same score of 1, whilst the lowest ranked documents across all sets will have a score of 0. These approaches can all be considered variants of the Borda method, however there are other rank transformations that can be explored.

3.5.2.2 Reciprocal Rank

Reciprocal rank is an alternative method of rank transformations that has been used in information retrieval (Ogilvie and Callan, 2003). Reciprocal rank is simply one over the current rank, formally given in Equation 3.7.

$$Norm_{score(x)} = \frac{1}{rank_x} \tag{3.7}$$

Unlike the previous transformations we have seen, this transformation is non-linear with respect to the initial ranking. Whilst the previous transformations always maintained the distribution that was present in the raw values being transformed (either scores or ranks), reciprocal rank introduces an exponential decay to the rank values, such that it highly weights documents which appear at the beginning of a ranked list versus those that appear further down. To visualise the difference between the Borda and reciprocal approaches, we took a list of 100 ranks ([1 : 100]), and applied both the Borda and reciprocal transformations, normalising the resulting scores to Z-Scores. The results are presented in Figure 3.4.

We can see clearly demonstrated the difference between the two rankings generated, with the Borda transformation presenting as a linear function, whilst the reciprocal approach generates a reciprocal distribution. The two graphs intersect at rank positions 5 and 60, demonstrating the aggressive weighting of the reciprocal approach for the top 5 ranked documents, with less decay in score given after rank 60. Clearly systems which implement reciprocal ranking transformations are seeking to bias combinations such that the top ranked results of each result list become heavily weighted.

These approaches, both score and rank transformations, allow the documents

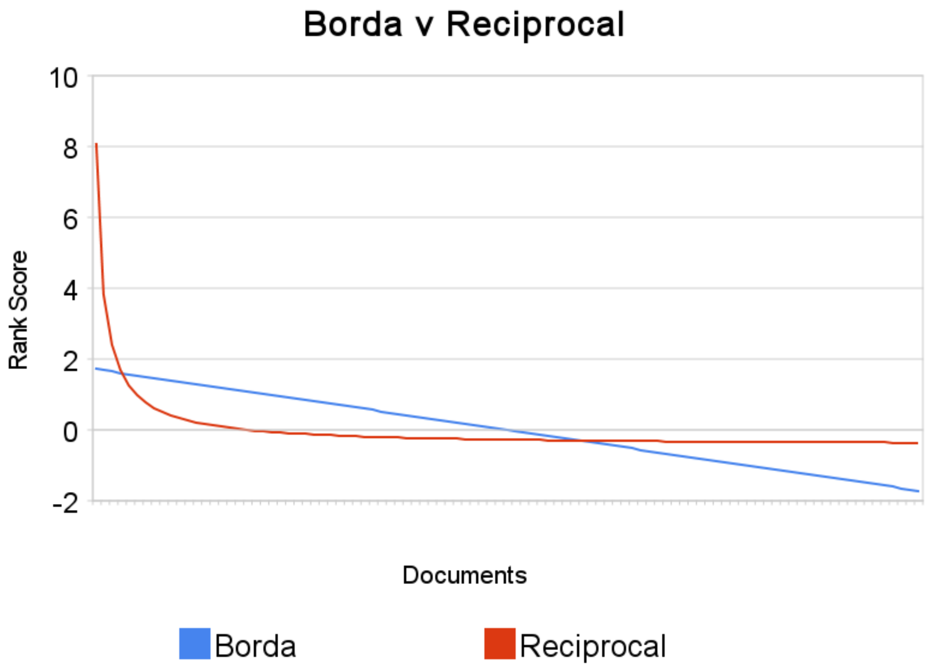


Figure 3.4: Borda vs. Reciprocal Rank Normalisation

from multiple ranked lists to be combined into a single response for an information need. The next section details approaches that can be employed for merging multiple ranked lists.

3.6 Combination Operators

Once the results from multiple ranked lists $rs_{i,j}$ have been normalised, they can then be combined to form a single ranked result for a given information need. In our discussion of the combination operators that can be used, we are assuming that one of the previously discussed transformations have been applied. If this is the case, then the combination approaches which we will now discuss can be applied to either the scores of a ranked list or to the rank of a ranked list. For convenience we will refer to a document's score when discussion combination approaches, however the normalised rank can be easily substituted and take the place of a score. The linear weighting of ranked lists and their subsequent combination has been proposed several times, with early work in linear weighting completed by Vogt and Cottrell (1999). A review of weighted data fusion approaches is presented in Chapter 5.

3.6.1 Round-Robin

Round-robin combination is one of the simplest methods that can be implemented and has seen use throughout the years (Savoy et al., 1996; McDonald and Smeaton, 2005). Given a set of ranked lists to be combined, take the top ranked document from each list and add it to the final list, then move onto the second ranked documents and so on, iterating through each of the lists. If a document is encountered multiple times it is assigned the highest rank it was given by any expert. Whilst this approach is easy to implement, and has some application in tasks of collection fusion where there may be little overlap in the documents being combined in ranked lists, its major drawback is in the introduction of randomness in the ranking. As this process takes documents from ranked lists, the order in which those lists are iterated through has a massive impact on performance with regards to metrics such as Average Precision. As no order is explicitly defined, this process of ranked list iteration is essentially random, and is the equivalent of a ranking where multiple scores which have the same value. Whilst documents with the same score are conceptually equal, as a ranking must be produced this forces an ordering of these equally scored documents which is random.

The more common approaches to combination of ranked lists is to use variants of linear interpolation. Fox and Shaw (1994) in early work defined six approaches for linear combination, of which two have seen the most success, CombSUM and CombMNZ. In defining their combination approaches Fox and Shaw first applied MinMax normalisation then combined. However this approach is unweighted, which has some justification for the combination of text retrieval experts (Lee, 1997; Beitzel et al., 2004), however as we have seen earlier in this chapter, the retrieval experts we are utilising in CBMIR are very noisy and any combination approach will require the use of weights in order to maximise performance.

Before explaining the combination approaches CombSUM and CombMNZ with weighting extensions, we will briefly recap our terminology. We have in our system the matrix of result sets (ranked lists) **RS** and we have a corresponding matrix of weights **RC**. Each ranked list $rs_{i,j}$ in **RS** has a corresponding weight $rc_{i,j}$ in **RC**. The task of any combination operator (such as CombSUM or CombMNZ) is to apply the weight $rc_{i,j}$ to the documents in the ranked list $rs_{i,j}$, and then fuse these weighted ranked lists into a final ranked list which forms the system's output for a query.

3.6.2 CombSUM

CombSUM is defined as the weighted sum of a documents score's in each of the result lists in which it appears. For instance, if we have our multi-part query Q and set of retrieval experts E, we will generate the matrix of result sets (ranked lists) **RS** and have along side it our weighting matrix **RC**. Given a document x, we examine each individual ranked list, $rs_{i,j}$, obtain the score of document x and apply the weight $rc_{i,j}$ to this score. The final score for document x therefore is the sum of each weighted instance of x occurring in the matrix **RS**. Formally this is described in Equation 3.8.

$$CombSUM(score_x) = \sum_{rs_{i,j}}^{RS} (score_x \in rs_{i,j}) \times rc_{i,j} \tag{3.8}$$

3.6.3 CombMNZ

An alternative implementation developed by Fox and Shaw (1994) is CombMNZ. CombMNZ extends CombSUM by introducing a variable which heavily weights documents that appear in more than one result set. This variable is the number of times that a document appears in result sets, i.e. the number of non-zero entries that a document has in the set of ranked lists to be combined (hence MNZ). In our terminology, this equates to the number of times a document x appears in a ranked

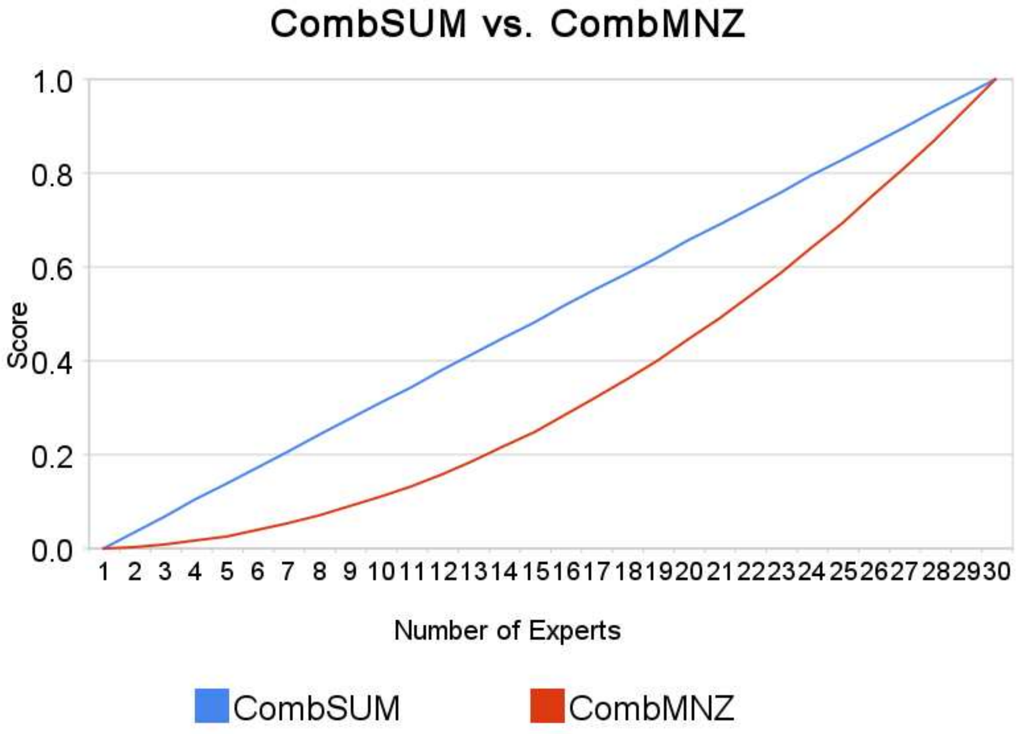


Figure 3.5: CombSUM vs. CombMNZ - hypothetical progression of scores of a single document

list ($rs_{i,j}$) in our result list matrix **RS**. We define weighted CombMNZ in Equation 3.9, where the weight has already been incorporated as part of the calculation of CombSUM.

$$CombMNZ(score_x) = CombSUM(score_x) \times \alpha \tag{3.9}$$

where α is the number of result sets in which a document was found, and $1 \leqslant \alpha \leqslant |\mathbf{RS}|$. Similar to some of the previous tools that we have examined, the choice of CombSUM versus CombMNZ is down to the intention of the system designer and the type of behaviour they would like present in the search system. CombMNZ aggressively promotes documents which occur in a majority of ranked lists to be combined, whereas CombSUM can be seen as a linear addition. We can visualise this with an hypothetical example shown in Figure 3.5.

Assume we have a set of 30 ranked lists which are being combined, listed on the

X-axis, and we have a document x which if found in a ranked list always receives the same score. In Figure 3.5 we plot the difference in the final score of document x as it is found in between 1 and 30 ranked lists. From this diagram we can see that with CombSUM, as more instances of document x are combined, that the final score of document x linearly increases. Conversely with CombMNZ we can observe a concave curve, which increases in gradient towards the end of the line, demonstrating the aggressive weighting of a document found in a majority of ranked lists, versus being found in only some of the ranked lists.

3.7 Combination Levels

As previously illustrated, the CBMIR system which we are using in our work is capable of handling multiple retrieval experts and multi-example multi-modality queries. We have seen that each unique pair $\langle expert_i, query_j \rangle$ is able to generate a ranked result set $rs_{i,j}$. The objective of the CBMIR system is to combine all of these into a coherent single response. The nature of this setup is that it allows for hierarchical partitioning of the ranked results to facilitate this final combination. That is, systems may employ a multi-stage combination process, either to allow for easier training of weights to populate the weighting matrix **RC**, or because employing a combination hierarchy was the accepted practice. For a late fusion CBMIR system, there are three basic levels of combination which could be employed, which we refer to as 'query level' combination, 'expert level' combination and 'direct level' combination. Each of these levels is illustrated in Figure 3.6.

Earlier in the definitions of our terminology, we stated that a CBMIR system could handle multi-example multi-modality queries Q. The set Q defined a single search topic which the CBMIR system is to process. The example as given in Figure 3.6, is that our query may be comprised of a yellow flower image and a red flower image, the intent of this query may be to find images of flowers regardless of colour (Jin and French, 2003). In the example in Figure 3.6 there are also two visual

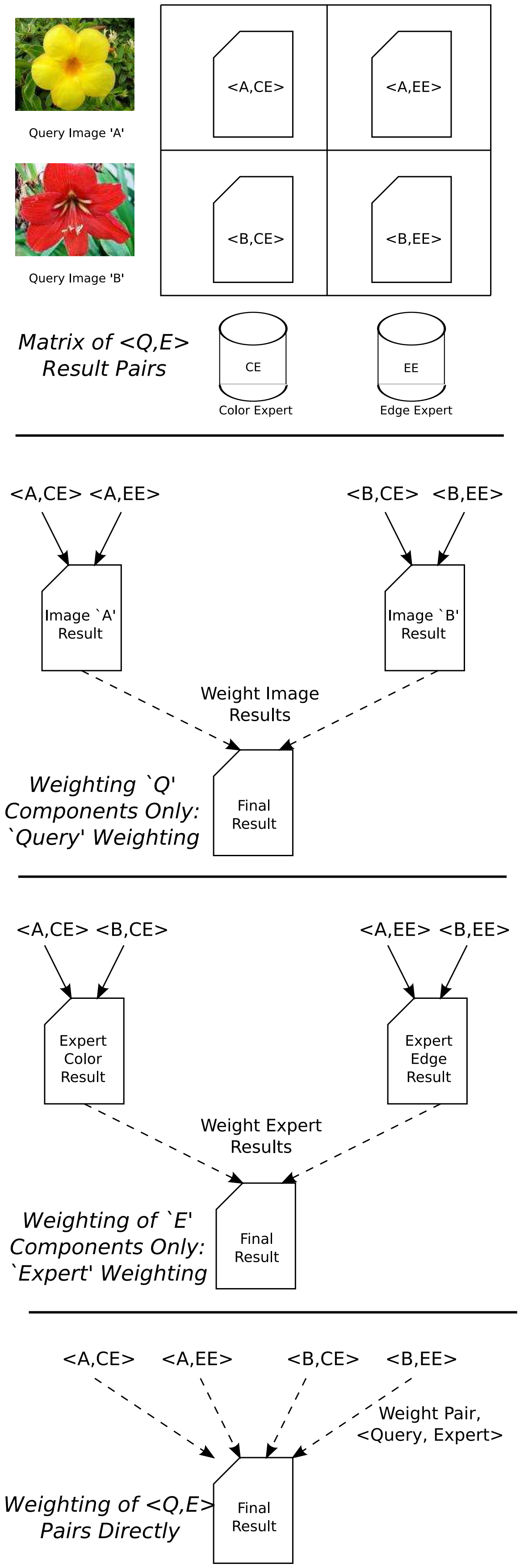


Figure 3.6: Combination levels for single search, with 2 experts (E) and 2 query images (Q), giving 4 ranked lists (pairs $\langle Q_j, E_i \rangle$).

experts present, a colour expert and an edge expert. Applying our queries to the experts generates 4 result lists to be combined. Using our notation the four result set we have generated are:

$$\begin{aligned}
E &= \{colour, edge\} \\
Q &= \{yellowflower, redflower\} \\
\mathbf{RS} &= \begin{pmatrix} colour, yellowflower & edge, yellowflower \\ colour, redflower & edge, redflower \end{pmatrix}
\end{aligned}$$

3.7.1 Expert Level Combination

The first of the three levels we'll discuss is 'expert level' combination. For a specific expert ($expert_i$) we query against it all query components in Q, merging the results to produce for each expert, a single ranked list. Typically the combination of the individual results from an expert into a single result for that expert are uniformly weighted. Therefore, for every $expert_i$ we have one combined result set (rs_i). The final merger therefore is to combine each result set rs_i into a single response. In systems which implement this style of combination, it is at this level of aggregation that weighting would occur, that is each rs_i would be assigned a weight. This means that the number of weights which must be determined using this level of aggregation is $|E|$. To illustrate this with the example from Figure 3.6, the result sets from pairs $\langle colour, yellowflower\rangle$, $\langle colour, redflower\rangle$ from one merged result set which is then weighted, whilst pairs $\langle edge, yellowflower\rangle$, $\langle edge, redflower\rangle$ form the other result set to be weighted. As there are two experts ($|E| = 2$), then two weights are used to calculate this query. Referring back to our result set matrix **RS** with individual elements $rs_{i,j}$, we can think of this approach as first aggregating the result sets in each column i, weighting each of these, then combining all instances of the columns rs_i into a single response.

3.7.2 Query Level Combination

If 'expert level' combination is the processing of the columns of the result set matrix **RS**, then 'query level' combination is the same processes except that we instead process the rows of matrix **RS**. Given a set of query components Q with members $query_j$, we combine with uniform weights the results of $query_j$ queried against every expert in E. Similarly to 'expert level' weighting, the aggregated result set for each $query_j$ is then weighted and combined to compute the final response to the query. Therefore, each rs_j is weighted, and the total number of weights used in tuning the system is $|Q|$. Illustrating this with our example from Figure 3.6, there are two result sets rs_j, the first generated from a merger of the pairs $\langle colour, yellowflower\rangle$, $\langle edge, yellowflower\rangle$, and the second from the pairs $\langle colour, redflower\rangle$, $\langle edge, redflower\rangle$.

3.7.3 Direct Level Combination

Finally we have the "direct" level of combination, where if the 'expert level' of combination was the aggregation and weighting of columns rs_i, and 'query level' combination was the aggregation and weighting of rows rs_j, then 'direct level' is the direct weighting of each individual result set $rs_{i,j}$, in other words, processing the matrix **RS** directly without any intermediate levels of aggregation. This level specifies weights for every coupling of a query component and retrieval expert, meaning that using this approach, $|E| \times |Q|$ weights are required. Applying this to our example from Figure 3.6, this means that the four pairs $\langle colour, yellowflower\rangle$, $\langle edge, yellowflower\rangle$, $\langle colour, redflower\rangle$ and $\langle edge, redflower\rangle$ each have their own weight.

In summary, these three levels of combination allow for different levels of granularity to be specified for combining results through data fusion. The more coarse combination levels are the 'expert level' allowing $|E|$ weights to be set, 'query level' allowing $|Q|$ weights and finally the 'direct level' of combination, which requires

$|E| \times |Q|$ weights to be specified. The selection of which level a system designer implements comes down to what mechanisms are available for estimating the weights to be used. For instance, if only coarse levels of training data are available, then the 'expert level' of combination may make sense. However what is clear is that the imposition of a combinatorial hierarchy will have a direct impact upon performance. In the next chapter as we examine the various factors which impact on data fusion we will demonstrate the degree to which the selection of combination levels has upon performance. We expect that the 'direct level' would perform the best, and what has currently not been shown in literature is how big of a performance impact is resultant from these design decisions.

3.8 Conclusion

In this chapter we have identified, isolated and defined factors which will impact upon performance in weighted data fusion. Many of these factors in previous studies have not have been explicitly identified and tested, such as read-depth, combination levels, normalization strategies and their interplay. Our intention here is to highlight the wide variety of factors which need to be taken account of when designing a data fusion framework for weighted combination, particularly within the context of CBMIR. In the following chapter we will perform rigorous experimentation with these various factors to determine what impact they may have upon data fusion. This will allow us to define what an ideal data fusion scheme should consist of and what factors require close attention. The factors we have identified are summarised in Table 3.2.

Summary of Data Fusion Factors	
Query & Experts	
	Single/Multiple Modality Queries
	Single/Multiple Retrieval Experts
Retrieval Factors	
	Sample rate of keyframe extraction (video)
	Read depth from each expert
Equivalence Transformations	
	Score normalisations:
	- MinMax, Z-Score
	Rank transformations:
	- Borda Count, Reciprocal Rank
Combination Operators	
	Weighted CombSUM
	Weighted CombMNZ
Combination Levels	
	Query-level
	Expert-level
	Direct-level

Table 3.2: Factors impacting on data fusion performance

Chapter 4

Evaluation of Factors Impacting on Multimedia Retrieval Performance

In this chapter we will be conducting a rigorous examination of the factors identified in the previous chapter which may impact upon retrieval performance in the context of Content-based multimedia information retrieval (CBMIR). CBMIR systems as a result of employing multiple low-level retrieval experts which are of poor individual quality are required to employ some form of a weighting scheme to combine this evidence so as to obtain a final result. This problem can be phrased as a *combination of experts* problem and is a case of *Data Fusion*. CBMIR is characterised by the use of multiple 'noisy' signals such as the colour of an image, or the textures it contains, and through combining multiple sources of noisy information, reasonable performance can be achieved Smeaton et al. (2006). However this makes the role of data fusion, particularly weighted data fusion, paramount to the success of many CBMIR systems.

Because of the employment of weighting schemes for combining evidence, the impact of various components within a retrieval system can become obfuscated, as the success or failure of any given set of weights will dominate retrieval performance.

Therefore there are two key components for the successful evaluation of different factors which may impact on retrieval performance. The first of these is the weighting employed itself, whilst the second are the additional factors such as how evidence is combined which are required to formulate a final response. The challenge is in creating a process where these two components can be disambiguated to allow for robust empirical testing of each part of a data fusion system.

To this end we utilise in this chapter an unconventional method of empirical testing where we conduct an optimisation of the retrieval process directly on the test data, so that we maximise the performance of the current set of variables under consideration. A complete description and justification for this is given in Section 4.1, but the key benefits of employing this empirical approach are:

1. We establish for any given query, what the ideal form of linear weighting is for that query. This allows us to identify and observe what form of weighting a data fusion algorithm should seek to emulate.

2. The capability of finding the ideal set of weights for any given query or set of retrieval factors allows us to essentially freeze the impact that the weighting scheme has on retrieval performance. Therefore we can robustly test factors such as combination operators, normalisation approaches and combination levels, where for each as the ideal set of weights has been used we can cross-compare results knowing that the performance achieved is the best performance possible using that particular combination of factors.

Cleverdon *et al.* remarked in some of the earliest robust empirical IR investigations, that the most important factors to be measured in the evaluation of information retrieval systems are recall and precision (Cleverdon et al., 1966). The experimentation we perform in this chapter will be cross compared using primarily the measure of average precision. Therefore the impact of each of the factors we will investigate will be assessed in terms of how it affects average precision.

The remainder of this chapter is organised as follows. We will first describe our experimental setup, including details of our experimental model and optimisation procedures. Next we will conduct our first optimisation operations and examine the weights which are generated so as to observe if there are any forms an ideal weighting scheme takes. Third, we will then begin our experimentation of the factors identified in the previous chapter, where we will demonstrate the several data fusion operators are not as effective as is generally believed in the data fusion community. Finally we will revisit the work of Lee (1997), who conducted some of the earliest investigations into data fusion performance, so as to investigate and contrast the findings we make in the chapter to what has been reported. The observations of this chapter will be utilized in Chapter 6 where we will develop our own novel algorithms for data fusion which will leverage the findings of this chapter.

4.1 Experimental Setup

In this section, we provide the motivation for our approach to examining the factors which impact on Content-Based Multimedia Information Retrieval. The context of our experimentation is an ad-hoc search task, where a system is given an expression of an information need and is required to return as many relevant matches as possible. For our investigation we performed 'fully automatic' retrieval which processes a query and produces a response with no human intervention.

Our experimental setup will detail the measures we use for examining the significance of our results, the experimental model which we will employ, the test corpora which will be used in the investigation and finally the optimisation model which forms the core of our experimental work.

4.1.1 Significance Testing

Testing for the significance of an experimental result is an important component of any empirical investigation. At its core, significance testing informs us of the probability of an empirical result occurring due to chance, or due to a deterministic process. An overview of significance testing and its application for Information Retrieval is given in Blanken et al. (2007). The significance test we will be utilising for our experiments is a partial randomisation test as implemented by NIST for significance tests in TRECVID. This approach has the benefit of being a non-parametric test which assumes no underlying distribution. Further details on the significance test are available from the TRECVID website[1].

Throughout our experimentation we will make extensive use of significance tests, where typically ρ will be set to 0.05. To present our significance results we will employ a consistent format, where the results are presented in a table, an example of which is given in Table 4.1. The top of the table indicates the test corpora and the ρ value. The table should be read left to right, not top to bottom, as the symbols used within the table convey direction. These symbols are; (1) $\equiv$, which means the two runs have no significant difference, (2) $\gg$ which indicates that the runs performance is significantly different and greater than what it is being compared to, and (3) $\ll$ which indicates that the run is significantly worse.

$\rho = 0.05$	TRECVID 2004			
	A	B	C	D
A	-	$\ll$	$\ll$	$\ll$
B	$\gg$	-	$\equiv$	$\ll$
C	$\gg$	$\equiv$	-	$\ll$
D	$\gg$	$\gg$	$\gg$	-

Table 4.1: Example Significance Table: $\equiv$ indicates runs have no significant difference, $\gg$ indicates the run is statistically better, $\ll$ indicates the run is worse, read from left to right, not top to bottom.

Using the example from Table 4.1, we can infer the following. Run ‘A’ is a poor performer, every other run out performs it. Run ‘B’ produced performance greater than ‘A’ but there was no significant difference between it and Run ‘C’, whilst run ‘D’ outperformed it.

[1] http://trecvid.nist.gov/

4.1.2 Experimental Model

The study of information retrieval, like any branch of science, has established various methodologies for advancing our body of knowledge. These techniques are implementations of the scientific method, where scientists make empirical observations about a body of data, formulate hypotheses and test these against the data in ways which can be accurately measured and crucially reproduced. For information retrieval, watershed moments were the Cranfield experiments and later the TREC series organised by NIST (Cleverdon et al., 1966; Harman, 1993). These events established many of the norms which can be taken for granted when conducting IR experiments, the establishment of common data collections, the specification of the experimental scenario (i.e. the topics) and the relevance assessments used to evaluate the performance of the search topics against the common data. The success of these experimental models has undoubtedly brought a great many advances to the field of information retrieval. Figure 4.1 demonstrates the traditional empirical IR model.

In this model we have some form of training data, either topics, data or both, some proposed model which we want to test and some form of parameters which require tuning and a set of evaluation metrics and relevance assessments. Also included in this model is a test set from which final results will be reported. The common sequence of events is that a model is first optimised on training data, then the optimised model is used on the test data. The final result typically reported is the outcome of the evaluation metrics run on the output of the model on the test data. There are several well founded justifications for employing this approach, most of which are concerned with overfitting the model and obtaining non-representative evaluation figures. The typical argument is that in a 'real' system, the system designers will not have access to the queries that will be used *a priori* but will have access to some form of training data. Therefore by training the model on the training data and executing that model with the unseen test data we have an objective measure of how well the model performs operationally on new search

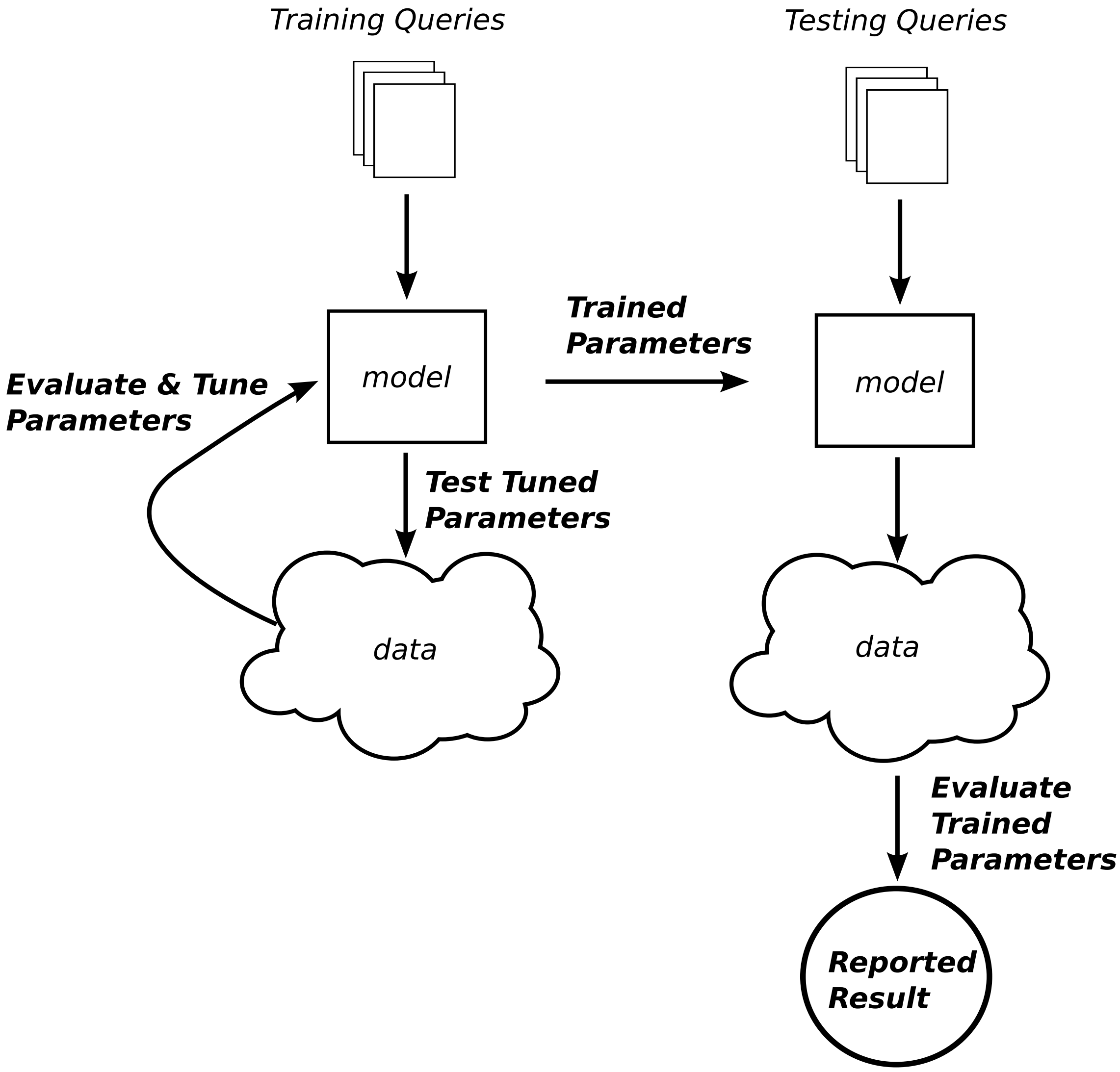


Figure 4.1: Traditional IR Empirical Model

queries. This approach is particularly useful when testing a model to see how well it generalises, training the model on one form of data then testing it on completely different test data, giving an indication if the model has general properties which make it widely applicable, or if the model has been too tailored to a particular type of data.

The application of this model however has become *de rigueur* amongst information retrieval systems researchers, with little thought as to why we are employing this experimental model and the validity of alternative approaches. It is seen as the 'correct' way to undertake a study, *ipso facto* deviations from this produce 'invalid' results as they break the experimental model.

The major problem with the established empirical model is that of evaluation, a problem which is more acute when applied to data fusion tasks. In data fusion tasks we will have a range of input sources of evidence, which we will then combine in some manner in order to compute a final response. The fundamental problem is that using the established empirical model, we can evaluate two different fusion models, and after executing both on the test collection we can make the observation that model 'a' outperforms model 'b' by 15%. On the surface this seems fine, model 'a' has achieved a good performance improvement over model 'b'. However, this 15% is a *relative* increase, it is only meaningful when comparing the two models under observation.

The problem is that we would like to know what the performance of each model is not with respect to each other, but against what the theoretical maximum performance achievable is, or in other words against the ideal data fusion combination. For instance, model 'a' scores a MAP of 0.115, model 'b' scores a MAP of 0.100, however if the theoretical maximum for the task is a MAP of 0.555, then the observed improvement whilst good indicates there is a lot more that could be done. Conversely if model 'a' scored a MAP of 0.523 and the maximum achievable remains 0.555, then we can make fairly good claims that model 'a' is achieving near peak performance and fewer performance gains can be expected.

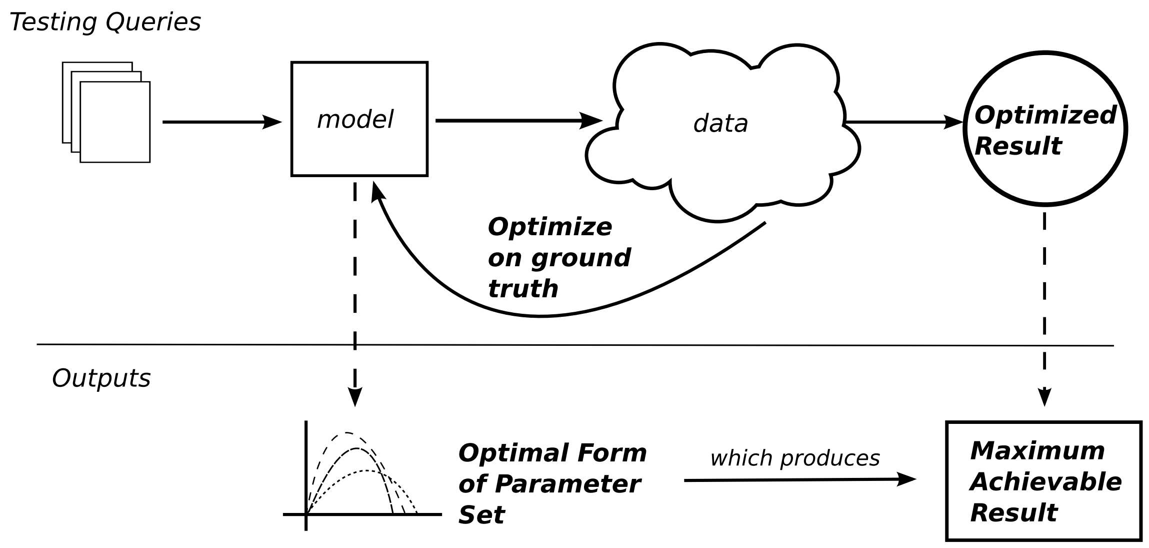


Figure 4.2: Optimisation on Test Data

A better use determining what the maximum performance may be, is to allow the study of what are the properties of a maximally performing model. Rather than being primarily concerned with what is the maximum performance value, to flip this around such that given we have a model which achieves excellent performance, what are the properties of this model that led to this performance. In order to achieve this it necessitates optimisation directly on the test data.

We present this alternative experimental model in Figure 4.2. There are two key characteristics of this model. Firstly that we only have one data set, the test data set along with relevance judgements, which enables the loop in the process to exist, allowing the tuning of the model sufficiently until its peak performance is reached. The second key characteristic of this model is in the outputs. Not only do we obtain a final ranked set of results for a given set of queries, but we also have from the model what parameters were used in order to obtain the peak performance.

Whilst this preamble justifying our approach is extensive, we feel it is necessary as we want to make clear our purpose in optimising retrieval performance directly on test collections, as this activity is quite unusual. By performing optimisation on the test set it allows us to identify what are the properties of an ideal weighting scheme for data fusion. Furthermore, as the optimisation generates the weights, it allows us to independently test other aspects of retrieval performance objectively, such as the effect of normalisation or combination operators, so that we can obtain strong observations as to the impact of these different methods.

All of the experiments presented within this chapter except where noted will make use of this experimental framework, directly optimising linear combination weights on test data. In so doing this will allow us to make robust observations as the variable of the impact of the linear weights utilised is always held constant. This constancy is achieved by those weights being set to what is the most effective set of weights for the variables under consideration. Nevertheless empirical testing of the type where the test collection is unknown is important for the development and testing of new algorithms. In Chapter 6 where we introduce new algorithms for the generation of weights for data fusion, we will employ the traditional empirical evaluation model.

At this point we would like to note that frequently in this Chapter, we will refer to an optimal run from which we compare the current factors under consideration. Due to the optimization process requiring significant processing time, we had to take best guesses as to what was the optimal set of factors to use to generate the best possible run for comparison. Completing this Chapter, we found that at times our guesses whilst close were at times incorrect (notably as we shall observe that rank based methods achieve superior performance to score based methods, contrary to accepted wisdom). Nevertheless, the optimal run for each experiment is directly comparable with the factor under consideration. Except where otherwise noted, all optimal runs utilized all available experts and query-components, CombSUM for combination, direct-level combination, score-based MinMax normalization and read-depth of 1000 documents. Deviations from this will be noted where applicable. In cases where one of these factors is being examined, then whilst that factor changed, the other variables would remain constant. E.g. for testing CombSUM against CombMNZ, except where noted, we used a read-depth of 1000 documents with direct-level of combination and score based normalization, whilst obviously

CombSUM and CombMNZ were interchanged.

Fundamental to implementing this experimental framework is the implementation of an appropriate optimisation framework. Several approaches were considered including standard grid searches and statistical approaches such as Expectation Maximisation. We selected the approach known as *coordinate ascent*. This approach was recently adapted for linear combination information retrieval tasks with direct optimisation on the relevance assessments by Metzler and Croft (2007). The following section will introduce this method and our extension to it.

4.1.3 Optimisation Technique and Extensions

To determine the optimal topic weights for all pairs $\langle Expert_i, Query_j \rangle$ in set $\mathbf{RC}$ we require an optimisation method which will directly maximise the evaluation function we are interested in, which for our purpose is Average Precision (AP). The method we use is an extension of *Coordinate Ascent* (also known as *Alternating Variables Method* (Fletcher, 1987)); its use in optimising Information Retrieval systems is described by Metzler and Croft (2007).

The overview of this approach is that to determine each topic's weighting matrix $\mathbf{RC}$, we first assign each pair $\langle Expert_i, Query_j \rangle$ a random weight (rc_{ij}), such that $rc_{ij} > 0$ and $\sum rc_{ij} = 1$. For each rc_{ij} we increment its assigned weight (whilst ensuring that the $\sum rc_{ij} = 1$; Metzler and Croft term this projecting the weights to a multinomial manifold (Metzler and Croft, 2007)) then apply the current weight set to each $\langle Expert_i, Query_j \rangle$ and re-evaluate against the evaluation function (AP). If there is an increase in AP then we continue to increment the current weight until AP no longer increases. Once the current value for rc_{ij} is optimised we move onto the next weight in the set. This process loops through $\mathbf{RC}$ successive times until further increments produce no performance increase. As commented by Metzler and Croft, in a multi-dimensional parameter space the evaluation function is unlikely to be concave in shape and the risk exists that we may finish on a local maximum. To alleviate this we instantiate this process with random weights multiple times, selecting as the optimal matrix **RC** which achieved the highest AP value.

Our extension of Coordinate Ascent introduces an extra step in the optimisation. The standard version of Coordinate Ascent terminates once no further increases can be made through incrementing the values of **RC**. In our approach, a second round of optimisation occurs, except that instead of incrementing each weight, we decrement each weight in the **RC**. We refer to this extension as Coordinate Ascent/Descent.

To demonstrate the advantage of using this extension, we compare the performance of runs optimised by standard Coordinate Ascent and our Coordinate Ascent/Descent. Both algorithms were executed with 50 random restarts on each of our corpora and we found that in all cases, Coordinate Ascent/Descent produced an optimised matrix **RC** which was statistically significantly better than the standard implementation (using a ρ value of 0.01). Conversely we found that if we used the standard implementation, but varied the amount of random restarts between 50 and 200, no significant difference could be found in the resulting outputs.

Our use of Coordinate Ascent differs to that of Metzler and Croft. The objective in their work was to test Coordinate Ascent as a method for determining the best parameter set to be used for ad-hoc retrieval from a training collection, then to apply those weights to a test collection and compare Coordinate Ascent to other training models, including SVMs. Their conclusions were that as Coordinate Ascent optimises directly on the evaluation metric (average precision) it produces superior weight sets for linear combination than the other methods they investigated (Metzler and Croft, 2007).

So as to demonstrate the effectiveness of the optimization process, in Table 4.2 we present the result of the optimization process run for each corpus with visual only experts, contrasted with a uniformed weighted run and the best reported automatic run from each of the corpora.

We show the comparison to the best reported runs in that year's evaluation, as

Eval.	MAP	Recall	P10	Uniform	BR
TV2003	0.1224	0.3027	0.3880	0.0593	N/A
TV2004	0.1084	0.2318	0.3826	0.0288	N/A
TV2005	0.1407	0.1725	0.6750	0.0646	0.1259
TV2006	0.0563	0.1493	0.4875	0.0164	0.0867
TV2007	0.1304	0.3231	0.6167	0.0422	0.0874
IC2007	0.2156	0.4379	0.5900	0.1283	0.1890*

Table 4.2: Optimised Results, column 'BR' is the best reported MAP for automatic search from that year's published results. 'Uniform' represents using all pairs $\langle Expert_i, Query_j \rangle$ but with no weighting. *IC2007 BR is visual only.

it demonstrates the effectiveness of our optimisation, producing retrieval runs which achieve excellent performance. The comparison highlights the maximum of what can be achieved with data fusion and global low-level visual experts, particularly when compared against the top performing runs which made use of multiple evidence modalities including text and semantic information. We note that this comparison to published retrieval runs ('BR') is not a fair comparison as we optimised on the test data, again however the intention of this work is to demonstrate the gains achievable with optimised weights, even when compared against retrieval runs that used high quality signals such as text.

4.1.4 Corpora Review

Whilst our experiments are making use of *coordinate ascent* for optimisation so as to isolate as many variables as possible in our investigation, one key aspect of variance is our actual test corpora. To determine if observed results are general or corpus specific we conduct our experiments over six test corpora. Five of these test corpora came from TRECVID (Smeaton et al., 2006) and are digital video collections, and one from ImageCLEF (Clough et al., 2008), a collection of travel photographs. These two campaigns share similar objectives as both seek to promote research in content-based retrieval by utilising common test collections and open, metrics-based evaluations. Within the five TRECVID corpora however we also have variation, with the data including mono and multilingual video, video from broadcast news and news magazine video. The following is a summary of our experimental corpora:

- **TRECVID 2003**: Approx. 60 hours of monolingual English news broadcasts. There are 72,624 total keyframes, of which 37,104 are 'NRKF' keyframes and 35220 are 'RKF' keyframes. There are 138 topic images spread across 25 topics. Text evidence is provided through ASR transcripts. Abbreviated to 'TV2003'.

- **TRECVID 2004**: Approx. 70 hours of monolingual English news broadcasts. There are 48,818 total keyframes, of which 33367 are 'RKF' keyframes and 15451 are 'NRKF' keyframes. There are 160 topic images across 24 topics, and text evidence is provided through ASR transcripts. Abbreviated to 'TV2004'.

- **TRECVID 2005**: Approx. 80 hours of trilingual news broadcasts in Arabic, Chinese and English, represented as 78,206 keyframes. Of these 45765 are 'RKF' keyframes and '32215' are 'NRKF' keyframes. Topics are represented by 228 topic images, across 24 topics. Text evidence is provided through ASR transcripts for English, whilst for the additional languages the ASR is run through an MT system. Abbreviated to 'TV2005'.

- **TRECVID 2006**: Approx. 160 hours of trilingual news broadcasts in Arabic, Chinese and English, represented as 146,497 keyframes. 'RKF' accounts for 79848 keyframes whilst there are 66844 'NRKF' keyframes. There are 169 topic images across 24 topics, text evidence is provided through ASR transcripts for English, whilst for the additional languages the ASR is run through an MT system. Abbreviated to 'TV2006'.

- **TRECVID 2007**: Approx. 50 hours of Dutch news magazine video, represented as 295,350 keyframes in total. Of these 19702 are 'RKF' images, whilst for this collection we took the aggressive sampling strategy of extracting 'K-

Frames', which make up the remaining 275648 images. For the topics there were 205 topic images across 24 topics. The audio for this video was nearly all Dutch, so all text was first detected by ASR which was then run through an automatic Machine Translation (MT) process. Abbreviated to 'TV2007'.

- **ImageCLEFPhoto 2007**: 20,000 natural still images which form the IAPR TC-12 Benchmark and 180 topic images across 60 topics. Text evidence comes from well-formed, noise free text annotations which accompany each image. Abbreviated to 'IC2007'.

In the remainder of this chapter we will be presenting an investigation into the variables identified in the previous chapter as to their impact upon weighted data fusion. In the following section we will first present an examination of the weights that are generated from the optimisation process, so as to determine what is the distribution of optimal weights for a given test corpus and if there are general patterns that can be observed in the ideal weighted set which may inform data fusion development.

4.2 Weighted Data Fusion

In this section we will be exploring the form of an ideal weighting scheme so as to determine if any unique properties exist within it which may guide future data fusion algorithm development. As previously stated our mechanism for doing this will be the execution of the coordinate ascent optimisation technique directly on our test corpora and relevance judgements. The first results of this process are presented in Figure 4.3, which is a histogram of the distribution of the ideal weights generated over all corpora. The y-axis represents frequency, whilst on the x-axis, the assigned weights have been transformed into Z-Scores, to allow for cross-comparison between topics and corpora. Again, Z-Score's shift and scale values to have a mean score of zero and standard deviation of one allowing us to express a value in terms of how

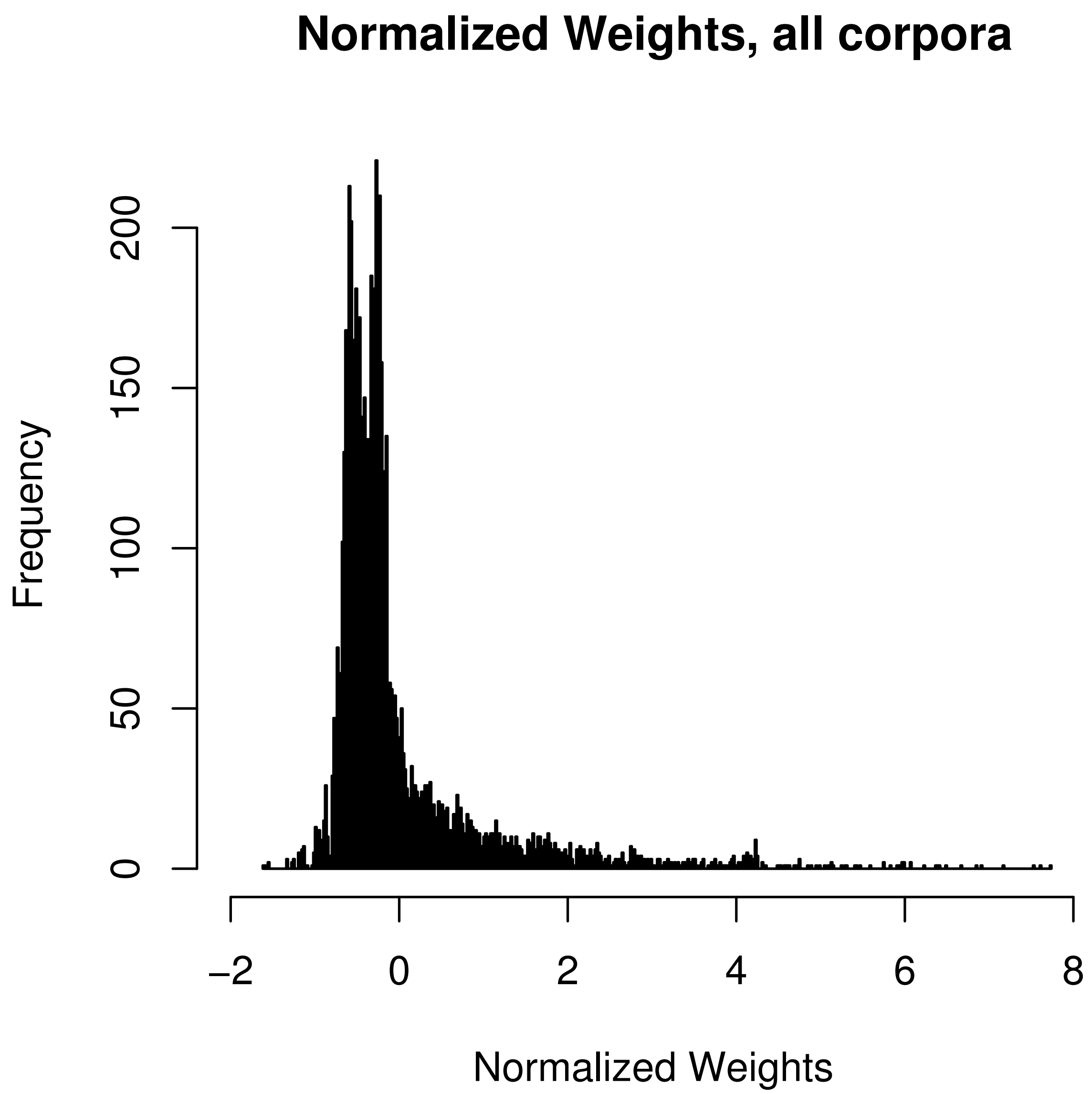


Figure 4.3: Histogram of weight distribution, all corpora

many standard deviations it is away from the mean. Therefore this normalisation allows us to examine how clustered assigned weights are. For clarity, each *topics* weights were normalized, meaning for each topic the average weight after normalization is zero. The results of the Z-Score's for each topic are aggregated into the presented graph and data.

From the histogram we can see demonstrated a highly positively skewed distribution, with a long tail of values extending up to Z-Score values of nearly 8σ. The shape of this distribution closely resembles that of a log-normal distribution, characterised by the extended positive tail. To further examine the skewness of the ideal weight distribution, we present various measures of central tendency in Table 4.3. From this we can see that whilst the mean of the weights is approximately zero and the standard deviation has a value of one, measurements that could be expected of a normal distribution after a Z-Score transformation, the median value is less than the mean value, correlating with our positive skew observation. Furthermore, an examination of the quantiles reveals a very high degree of clustering with an extended tail, as the 75^{th} quantile has a value of only 0.0036, whilst it isn't until the 90^{th} quantile that values exceed one.

Mean	Median	σ	75 Quantile	90 Quantile
0.0004	-0.2978	1.0004	0.0036	1.1192

Table 4.3: Measures of central tendency for ideal weight distribution, all corpora.

We can infer multiple insights from the presented distribution and measures of central tendency. Firstly, that whilst the distribution of weights has some properties of that of a normal distribution, such as a majority of the data points clustered around the mean and within the range $\pm 3\sigma$, there does exist a very definitive positive skew. Secondly, as part of this positive skew approximately 10%-11% of the weights were assigned values $> 1\sigma$. The implications of this are that overall the initial observations would suggest that a minority of the pairs $\langle Expert_i, Query_j \rangle$ received the majority of a topic's weight.

The histogram presented is an amalgamation of the weights for all topics over all corpora. Without other evidence there remains the possibility that the effect presented is a corpora-specific event and that the weights are indeed more normally distributed. To account for this we present in Figure 4.4 corpora-specific plots of the weight distributions in the form of quantile-quantile (Q-Q) plots. In each of these figures, the x-axis represents a theoretical normal distribution of weights, whilst the y-axis is the actual weight which was assigned. The dashed line displays the trend line of the weights if they were normally distributed.

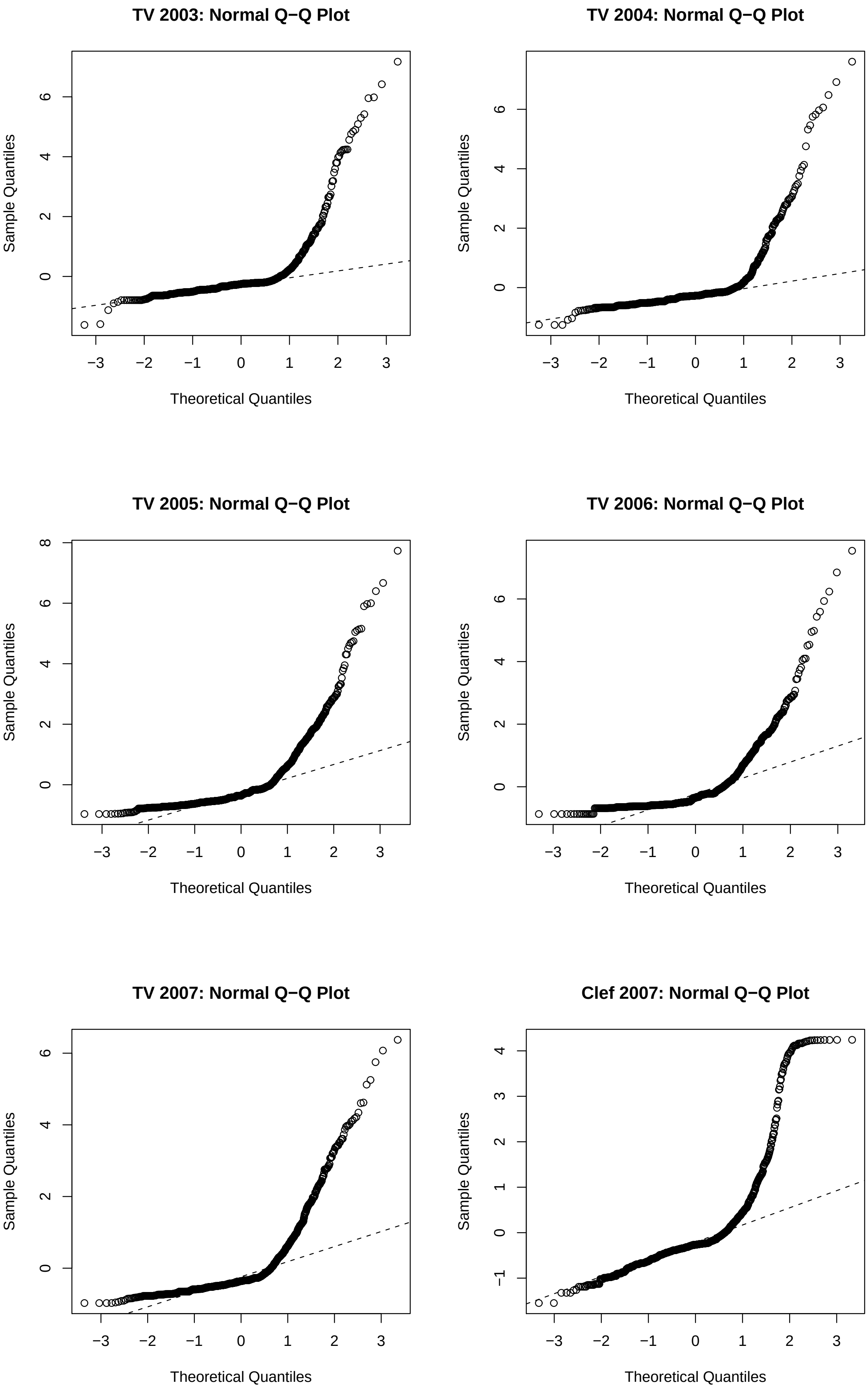


Figure 4.4: Q-Q Plots of Weight Distributions

Examining each of the six Q-Q plots, we can see that all of our experimental corpora follow the same distributional pattern, as each demonstrates a significant departure from a normal distribution, particularly once the normalised weights values exceed 1σ. The pattern shown in each plot is similar to what would be expected if the distribution of the weights was log-normal, again we can also see demonstrated in each plot a positive skew.

Therefore, the evidence presented suggests that from the optimisation process, the ideal weighting form for data fusion constitutes a majority of pairs $\langle Expert_i, Query_j \rangle$ being assigned relatively low weights, whilst a handful of select pairs being aggressively weighted. Once again there remains the possibility that this observation whilst corpora-independent may be topic dependent. To explore this, we examined the distribution of weights within each topic for each corpora, the results of which are presented in Figure 4.5.

In this figure there is a graph for each of our experimental corpora. These graphs were constructed to examine for each topic, how many of the pairs $\langle Expert_i, Query_j \rangle$ were assigned a high weight, and of the total weight for a topic, how much of the weight did these highly weighted pairs account for. Therefore we examined for every topic what proportion of pairs $\langle Expert_i, Query_j \rangle$ were assigned normalised weights greater that 1σ, and for these highly-weighted pairs, what the sum of their weights was, or in other terms what proportion of the total weight did these highly-weighted pairs constitute.

Within each graph the x-axis represents individual topics for each corpora. For every topic there are two bars, a yellow bar which represents the proportion of pairs $\langle Expert_i, Query_j \rangle$ which received weight greater that 1σ, and a blue bar which demonstrates what percentage of the total weighting did that represent. Taking as an example the first topic in corpora TRECVID 2003, we can see a yellow bar at approximately 16% and a blue bar at approximately 78%. This indicates that for topic '0100' in TRECVID 2003, 16% of the pairs $\langle Expert_i, Query_j \rangle$ used for that topic received 78% of the total weighting. From these graphs we can see that

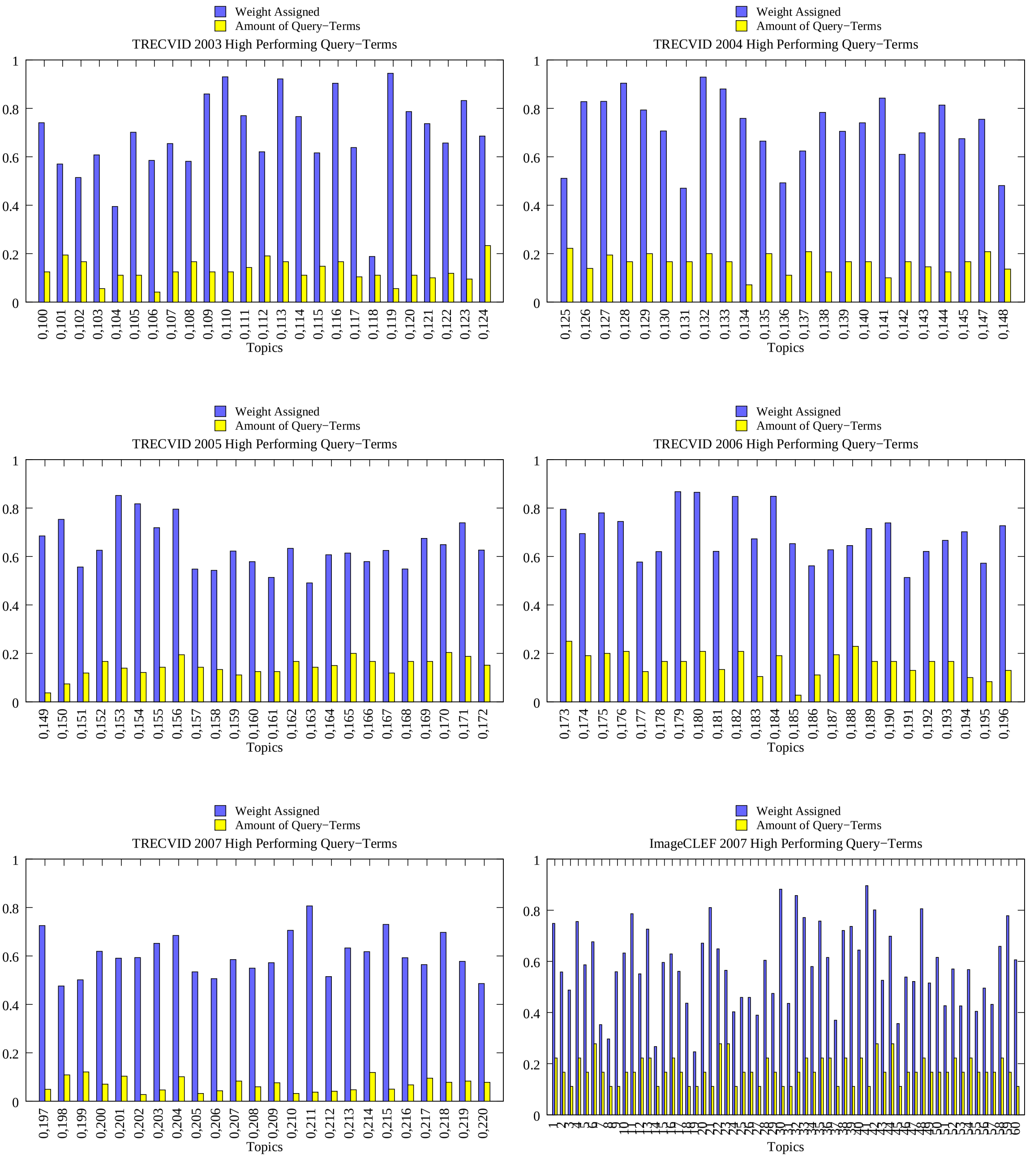


Figure 4.5: Highly Weighted Pairs, all corpora

generally across topics the pattern remains the same, that between 10%-20% of the pairs $\langle Expert_i, Query_j \rangle$ used for that topic attracted between 60%-80% of the weight. Unlike our previous observations, these graphs were also generated from an optimisation process which utilised visual only experts. This was selected so as to further determine if it was just one modality, i.e. text, having a significant impact upon the distributions. From these results we can see that the general effect holds, that a minority of pairs attract a majority of the weight. The possibility does exist that even in this case, it may be one particular visual expert which is dominating the weighting and thereby causing the skewed weighting distribution. Table 4.4 shows the break down of visual experts and the proportion of the weight assigned. There is a slight bias towards the Edge Histogram and to a lesser extent the Homogeneous Texture experts, however as there are only two texture but four colour experts, this bias can be accounted for. Taken together, the data presented in Figure 4.5 and Table 4.4 shows that highly weighted $\langle Expert_i, Query_j \rangle$ are distributed across different experts.

CL	CM	CS	SC	EH	HT
15%	17%	13%	13%	24%	18%

Table 4.4: Distribution of Retrieval Experts in $\langle Expert_i, Query_j \rangle$ with $rc_{i,j} > 1\sigma$

We have observed that the key to maximising AP is to correctly identify salient pairs $\langle Expert_i, Query_j \rangle$ and ensure that these are highly weighted, rather than weighting the overall performance of any given retrieval expert. Therefore the task now is to test this observation to determine how robust it is. We have devised a series of experiments that utilise only highly weighted pairs $\langle Expert_i, Query_j \rangle$ to see if we still achieve good performance, or if the remaining lowly-weighted pairs are contributing to maximising performance. For clarification, a highly-weighed pair $\langle Expert_i, Query_j \rangle$ is a pair whose weight rc_{ij} is greater than $+1\sigma$ of the mean weight for that topic.

4.2.1 Highly Weighted Pairs Experimentation

To test our observations we devised four experiments in order to (1) determine to what extent the highly-weighted pairs $\langle Expert_i, Query_j \rangle$ impact upon performance; (2) to determine if the weighting of these pairs needs to be exact or if merely identification is enough; and finally, (3) to determine the impact the remainder of the pairs $\langle Expert_i, Query_j \rangle$ which do not have much weight allocated to them have upon performance. The conditions of this experiment are that visual only experts were utilised, with CombSUM for combination and MinMax score normalisation. The four experiments we defined to test our observations are as follows:

- **(1σ) 1σ**: For each topic, only use highly-weighted pairs $\langle Expert_i, Query_j \rangle$ (i.e. pairs $\langle Expert_i, Query_j \rangle$ whose assigned value from optimisation was $+1\sigma$ for the mean weight). The value of w_{ij} will be the value determined during optimisation (Section 4.1.3). This test will examine the impact of precisely weighted high-performing pairs $\langle Expert_i, Query_j \rangle$. It can be thought of as a high-precision experiment as for each topic we will be using only 5%-20% of the available ranked lists for that topic.

- **(1σU) 1σ Uniform**: Using only the highly-weighed pairs $\langle Expert_i, Query_j \rangle$, assign each a uniform weight. This will examine if just the identification of high-performing pairs $\langle Expert_i, Query_j \rangle$ is sufficient to yield performance increases, specifically determining if accurate weighting of pairs is required, or if they can be assigned a binary weight [0,1]. As the task of determining the optimal set w_{ij} is realistically only viable post-experiment, this experiment tests if realistic fusion approaches can be developed, as it does not require perfect weights, only identification of likely high performing pairs $\langle Expert_i, Query_j \rangle$.

- **(1σU-T) 1σ & Tail**: We extend experiment 1σ, by taking the remaining weight mass that isn't assigned to high-performing pairs and allocate it uniformly amongst the remaining pairs in **RS**. This experiment complements the previous, we assign a large weight to the high-performing pairs, whilst a low weight to the remainder. As the high-performing pairs constitute only 5%-20% of available pairs for a topic, this experiment is testing the impact of recall, i.e. can we include the remainder of the data without accurate weighting so as to increase our recall.

- **(1σU-T) 1σ Uniform & Tail**: As above we find each of the high performing pairs then taking the weight assigned we give each a uniform weight from the high-performing weight mass. The remainder of the weight mass is then equally assigned amongst the remaining pairs. For example, if we have 4 pairs, whose optimised weights were 0.5, 0.4, 0.02 and 0.08, the high performing pairs would get ((0.5 + 0.4)/2) weight each, i.e. 0.45, and the last two pairs would each receive 0.05 weight.

The results of this experiment are presented in Figure 4.6 for each of our six corpora, with significance data given in Figure 4.7 and finally the results graphed in Figure 4.8. To provide context for the presented results we also include the runs *uniform* and *optimised* which provide a lower and upper bound on performance, where the *uniform* run has no weighting employed, whilst the *optimised* run is the result of all pairs $\langle Expert_i, Query_j \rangle$ being assigned their ideal weights. Next to each of our runs we include in brackets how close that run came to achieving the performance of the optimised run.

Generally from these results we can see that the run 1σU-T is the best performer, although this is not an unexpected result as it is the closest weighting form to the fully optimised weights. This run usually achieved a significant difference to all other runs. A notable exception to this is the TRECVID 2004 corpora, where none of the executed runs achieved a significant difference to each other. The inclusion of the 'tail' pairs assigned the remainder of the weighting mass, never significantly hurt performance, which is to be expected as primarily we should expect them to assist in improving recall. Generally these lowly-weighted pairs helped to improve performance, indicating that whilst they are not a significant driver of retrieval

Legend	TRECVID 2003		TRECVID 2004	
Run	**MAP**	**Recall**	**MAP**	**Recall**
Uniform All	0.0593	0.2375	0.0288	0.1440
1σ Uniform	0.0966 (79%)	0.2786	0.0738 (68%)	0.2268
1σ Uniform & Tail	0.0958 (78%)	0.2829	0.0764 (71%)	0.2251
1σ	0.0989 (80%)	0.2805	0.0770 (71%)	0.2246
1σ & Tail	0.1025 (83%)	0.2852	0.0805 (74%)	0.2229
All Optimised	0.1224	0.3027	0.1084	0.2318

Legend	TRECVID 2005		TRECVID 2006	
Run	**MAP**	**Recall**	**MAP**	**Recall**
Uniform All	0.0646	0.1140	0.0164	0.0926
1σ Uniform	0.1037 (74%)	0.1484	0.0460 (82%)	0.1332
1σ Uniform & Tail	0.1109 (79%)	0.1513	0.0453 (80%)	0.1379
1σ	0.1108 (79%)	0.1574	0.0496 (88%)	0.1393
1σ & Tail	0.1198 (85%)	0.1600	0.0498 (88%)	0.1409
All Optimised	0.1407	0.1725	0.0563	0.1493

Legend	TRECVID 2007		ImageCLEF 2007	
Run	**MAP**	**Recall**	**MAP**	**Recall**
Uniform All	0.0422	0.2007	0.1283	0.4095
1σ Uniform	0.0701 (54%)	0.2853	0.1404 (65%)	0.3809
1σ Uniform & Tail	0.0742 (57%)	0.2961	0.1715 (80%)	0.4128
1σ	0.0862 (66%)	0.2895	0.1439 (68%)	0.3814
1σ & Tail	0.1011 (78%)	0.3000	0.1755 (81%)	0.4172
All Optimised	0.1304	0.3231	0.2156	0.4379

Figure 4.6: Highly Weighted Experimental Results, all corpora.

$\rho = 0.05$		TV2003				TV2004			
Legend		1σU	1σU-T	1σ	1σ-T	1σU	1σU-T	1σ	1σ-T
1σ Uniform	(1σU)	-	$\equiv$	$\equiv$	$\ll$	-	$\equiv$	$\equiv$	$\equiv$
1σ Uni. & Tail	(1σU-T)	$\equiv$	-	$\equiv$	$\ll$	$\equiv$	-	$\equiv$	$\equiv$
1σ	(1σ)	$\equiv$	$\equiv$	-	$\equiv$	$\equiv$	$\equiv$	-	$\equiv$
1σ & Tail	(1σ-T)	$\gg$	$\gg$	$\equiv$	-	$\equiv$	$\equiv$	$\equiv$	-

$\rho = 0.05$		TV2005				TV2006			
Legend		1σU	1σU-T	1σ	1σ-T	1σU	1σU-T	1σ	1σ-T
1σ Uniform	(1σU)	-	$\ll$	$\ll$	$\ll$	-	$\equiv$	$\ll$	$\ll$
1σ Uni. & Tail	(1σU-T)	$\gg$	-	$\equiv$	$\ll$	$\equiv$	-	$\ll$	$\ll$
1σ	(1σ)	$\gg$	$\equiv$	-	$\ll$	$\gg$	$\gg$	-	$\equiv$
1σ & Tail	(1σ-T)	$\gg$	$\gg$	$\gg$	-	$\gg$	$\gg$	$\equiv$	-

$\rho = 0.05$		TV2007				IC2007			
Legend		1σU	1σU-T	1σ	1σ-T	1σU	1σU-T	1σ	1σ-T
1σ Uniform	(1σU)	-	$\equiv$	$\ll$	$\ll$	-	$\ll$	$\ll$	$\ll$
1σ Uni. & Tail	(1σU-T)	$\equiv$	-	$\ll$	$\ll$	$\gg$	-	$\gg$	$\ll$
1σ	(1σ)	$\gg$	$\gg$	-	$\ll$	$\gg$	$\ll$	-	$\ll$
1σ & Tail	(1σ-T)	$\gg$	$\gg$	$\gg$	-	$\gg$	$\gg$	$\gg$	-

Figure 4.7: Significance Testing, Highly Weighted Pairs

performance, their inclusion if assigned an appropriately low weight can boost recall.

The ImageCLEF 2007 corpora however does stand out as behaving differently to the TRECVID corpora, notably by the performance discrepancy between the ‘tail’ and non-tail runs. This indicates that recall plays a significant part in driving performance for this corpora, and therefore the inclusion of all pairs, no matter how weighted, contributes significantly to performance. This is highlighted by the relatively high performance achieved by the ‘uniform’ run for ImageCLEF 2007 where no weighting is applied, where the distance in performance between the ‘uniform’ and ‘optimised’ runs is the closest of any of our test corpora. This is particularly noticeable when the recall level is examined, with the difference in recall being less than 8%. The ImageCLEF 2007 corpora is the smallest of our test corpora, and in comparison to the TRECVID collections is certainly the most ‘noise free’ with high quality, non-redundant images and good quality textual data.

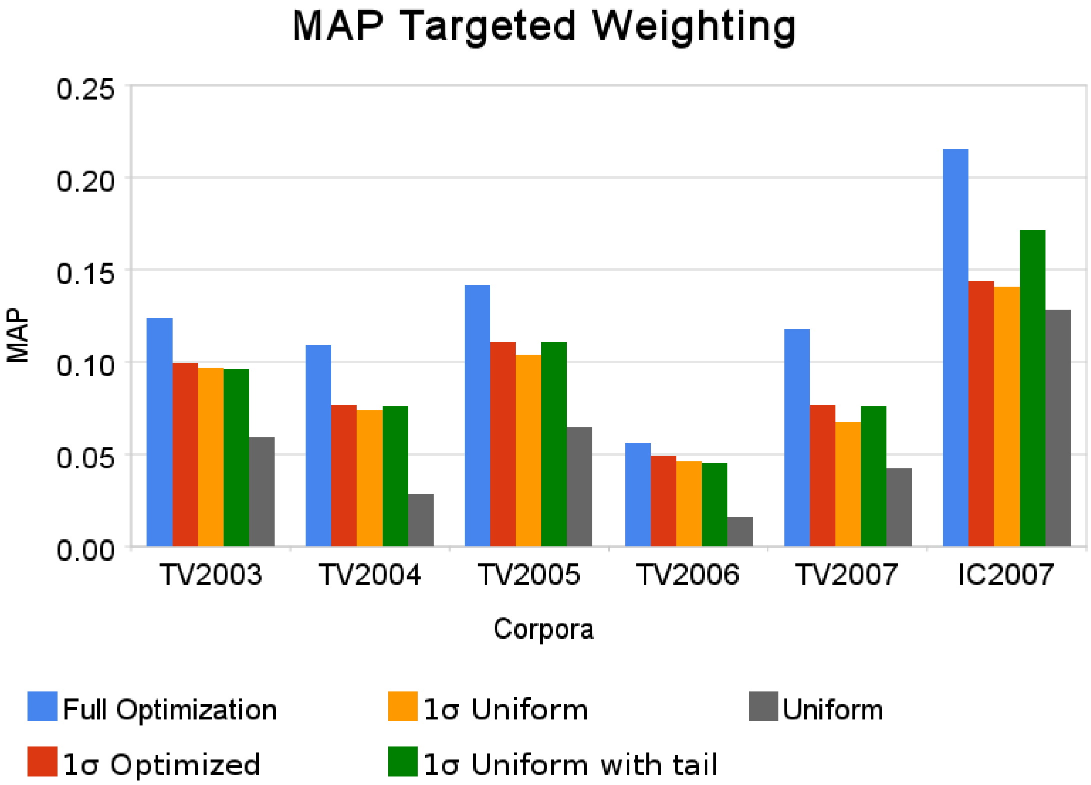


Figure 4.8: Targeted Weighting Performance

The results of these experiments clearly demonstrate the impact of correctly identifying the highly-weighted pairs, as they provide a massive impact in terms of driving retrieval performance. The run 1σ highlights that when using a subset of pairs $\langle Expert_i, Query_j \rangle$ from **RS**, very good performance can be achieved despite a reduction in potential recall by not using all pairs. Far more encouraging is the performance of runs 1σU and 1σU-T, which on average achieved 70% and 74% respectively of the fully optimised run performance. Whilst run 1σ had value as an illustrative run, it is hard to conceptualise a data fusion algorithm that would create the exact optimal weights for these pairs *a priori*. The runs 1σU and 1σU-T however did not use the optimal weights, but rather only identified from the matrix **RS** which were the high-performing pairs. As 1σU and 1σU-T were essentially employing a binary weighting scheme yet still achieved excellent performance, it provides a clear direction for development of data fusion algorithms.

The fundamental finding of these experiments nevertheless remains that topic-specific pairs of query components and retrieval experts ($\langle Expert_i, Query_j \rangle$) are the key to obtaining maximal performance for low-level data fusion. The identification of these pairs will lead to demonstrable large increases in retrieval performance. This can only be obtain by the employment of query specific approaches, but the increase in performance dictates that investigations into approaches which can exploit and identify these pairs will provide the next leap in CBMIR performance where low-level features are used.

4.2.2 Weighted Data Fusion and Overlaps

A key class of metrics used in examining data fusion was the overlap metrics, defined by Lee (1997) which measured either how many relevant documents did result sets share, or how many non-relevant documents were shared. One of the major outcomes of Lee's work was the definition that data fusion performance was driven by these overlaps, that result sets when being combined would share large sets of relevant documents but share few non-relevant documents.

The measure of overlaps which we defined in the previous chapter are not directly applicable to our task of examining overlaps in multimedia retrieval. As we have previously noted, the retrieval experts used for CBMIR are very noisy, particularly when compared to text retrieval counterparts. Because of this, the standard overlap measures fail, both $R_{overlap}$ and $NR_{overlap}$ as in both cases there are no documents which are common across all of the result sets being compared, meaning the overlap coefficients are always zero. Furthermore, in the work of both Lee (1997) and Beitzel et al. (2004) who have used the overlap metrics, they examined the overlaps from the context of an entire retrieval run, whereas for our examination we will be examining the overlaps at the topic level.

Therefore for our examination we modify the overlap formulas such that for each individual set rs_i, j we calculate the overlap of its documents versus the remaining result sets in **RS**. We then complete this for every result set, meaning that for each result set we have $R_{overlap}$ and $NR_{overlap}$ values. That means for every topic, we now have a set of $R_{overlap}$ and $NR_{overlap}$ values, which allows us to take the average of these so as to derive an average topic overlap measure.

By doing this we are left for each topic, in each corpora, average measures of $R_{overlap}$ and $NR_{overlap}$. This allows us to examine the data fusion hypothesis by examining the correlation of the overlap measures with average precision, to determine if there is a relationship between the two for CBMIR. There are two correlations we wish to examine, both of which utilise these topic dependent measures. Following the work of both Lee and Beitzel, the first correlation we will examine is the correlation of average precision with the ratio of $\frac{R_{overlap}}{NR_{overlap}}$ (abbreviated to R/NR). The higher the value of this ratio, the greater the difference between the measures $R_{overlap}$ and $NR_{overlap}$. A score of one for this ratio indicates that both $R_{overlap}$ and $NR_{overlap}$ are in equal proportion, greater than one that $R_{overlap}$ dominates and conversely $NR_{overlap}$. The second correlation we will examine is the relationship between average precision and $R_{overlap}$ only. For both of these tests we examine the correlation on optimised runs where we utilised both visual only experts and

	AP & Ratio R/NR		AP & $R_{Overlap}$		Avg. R/NR
Year	Image Only	Text/Image	Image Only	Text/Image	Text/Image
TV 2003	0.88	0.75	0.77	0.66	2.609
TV 2004	0.87	0.79	0.67	0.64	1.956
TV 2005	0.90	0.88	0.90	0.88	2.594
TV 2006	0.91	0.88	0.86	0.81	2.020
TV 2007	0.92	0.92	0.91	0.90	2.321
IC 2007	0.93	0.70	0.82	0.60	2.987

Table 4.5: Correlation of AP to Overlap measures.

visual and text experts. The results of these tests are presented in Table 4.5, the correlation statistic used was Pearson correlation measure.

From the presented results we observe a strong correlation between AP and R/NR, with a positive correlation also existing between AP and $R_{overlap}$ although this is slightly weaker. The high correlation between AP and R/NR indicates that good retrieval performance was gained when the R/NR ratio was maximised. This fits with the data fusion hypothesis, as it indicates that when multiple result sets returned similar relevant documents but dissimilar non-relevant documents that performance in terms of AP was maximised.

In the rightmost column of Table 4.5 we show the average value of R/NR over each of our testing corpora. This figure is quite large, as it indicates that on average as a result of our optimisation process, there were twice as many relevant documents found in common as there were non-relevant documents. This proportion is quite high, particularly when compared to the overlap statistics reported by Beitzel et al. (2004), where for text IR data fusion, the overlap ratios being reported were demonstrating a difference of between 20% and 25% for their test corpora within the same system, and 50%-70% for overlaps between different systems.

One of the main outcomes of the work of Beitzel *et al.* was demonstrating that for highly effective retrieval systems, such as text retrieval, data fusion with CombMNZ and the use of R/NR ratio's for determining the potential success of data fusion were *not* effective as highly effective systems already returned relevant documents in highly ranked positions, therefore often documents promoted through data fusion up a ranked list were common non-relevant documents. The result we present here of the strong correlation between AP and R/NR for linear weighted data fusion in CBMIR highlights the differences between data fusion utilising highly effective retrieval systems, as is the case in many text IR systems, versus using poor individual retrieval systems which are employed by CBMIR systems. This indicates that the behaviour of data fusion when studied within the domain of text IR may not be directly applicable to data fusion within CBMIR, as the elements being fused together within CBMIR have different properties which distinguish themselves from text IR systems, notably that raw performance is much worse and necessitates the use of weighted data fusion.

We would also note that whilst we have been observing that on the whole, low-level retrieval experts are particularly noisy sources of information, especially when evaluated by MAP, that for specific topics, various retrieval experts can attain good levels of performance. In particular, different experts will present different, useful, sources of information for the query being processed. As such this goes to the heart of the data fusion problem, where we have sources of information which vary greatly in quality, and thus we require a topic dependent weighting in order to properly combine these sources such that we achieve good retrieval performance. The data presented in this section highlights that low-level features do in fact return relevant information, just that it is very topic dependent which expert with which query component will perform well, and as such this high level of variability distinguishes the problem from text retrieval empirical observations.

Therefore we have observed a degree of difference between the observed effects of data fusion from text IR approaches and CBMIR data fusion tasks. In the remainder of this chapter, we will systematically test the data fusion variables identified in the previous chapter. For each set of variables examined we will be utilising the optimisation process defined earlier in this chapter. As a result, in each of the upcoming experiments the weights utilised are close to the optimal for that particular set of conditions under examination. This will allow us to make robust observations of the variables under consideration.

4.3 Retrieval Factors

In this section we will examine two 'retrieval factors' which we identified in the previous chapter, namely the effect of increasing the read depth and the number of experts we use for retrieval. Read depth is the number of documents we request from an expert for combination. Traditionally this has been 1,000 documents, the level that TREC requests participants to provide results for. We would note the distinction between increasing read depth and increasing the amount of results returned as the final response to a query. In these experiments we are altering the number of documents that are read from an expert for the purposes of data fusion, however the number of documents returned for a query is still fixed to 1,000. Therefore our evaluation metrics should be cross-comparable to other results as we are not returning more documents for evaluation. The second variable we are testing is the impact of adding in more retrieval experts for a given query, to determine the impact of adding in additional sources of evidence.

For both of these experiments we utilised only visual retrieval experts. We did this because for several of the topics across various corpora, the text expert returned far less than 1,000 results. For our read depth experiments we are examining reading amounts of 500, 1000, 3000, 5000 and 10,000 documents from each expert, whilst again restricting the final result to 1,000 documents. In all experiments, we used CombSUM and MinMax normalisation along with our optimisation process to obtain our results.

Our first experiment combines a read depth examination with varying the number of experts used. It is an optimisation of TRECVID 2005 topic '0165 - Find shots of basketball players on the court'. The results of this experiment are presented in Table 4.6 and visualized in Figure 4.9.

From this data we can firstly observe that adding in additional experts, partic-

	Read Depth				
Experts	500	1000	3000	5000	10000
1	0.0786	0.0784	0.0881	0.0919	0.0854
2	0.1221	0.1346	0.1450	0.1478	0.1460
3	0.1479	0.1755	0.2199	0.2324	0.2241
4	0.1525	0.1754	0.2168	0.2294	0.2163
5	0.2386	0.2834	0.3199	0.3215	0.3123
6	0.2502	0.2975	0.3305	0.3307	0.3178

	Relative Changes				
Experts	500	1000	3000	5000	10000
1	0%	0%	12%	17%	9%
2	0%	10%	19%	21%	20%
3	0%	19%	49%	57%	51%
4	0%	15%	42%	50%	42%
5	0%	19%	34%	35%	31%
6	0%	19%	32%	32%	27%

Table 4.6: Read Depth and Expert Variation, Topic 0165

ularly when optimised, increases retrieval performance. Whilst hardly a surprising outcome, it reinforces earlier observations about the ability of data fusion to enhance retrieval performance when we contrast these figures to the results that single retrieval experts achieve alone. Secondly, we can generally see that significant performance increases are actually obtained up to read depths of 3,000 documents per expert. After this point the metrics begin to saturate and either stabilise or even deteriorate slightly. There is an outside chance that this is a topic effect, rather than general behaviour. We would note that clearly there is an implicit ordering in the addition of the experts in this experiment. We knew a priori the performance of each expert, and as such we added the experts together in performance from best to worst. Nevertheless, we find that we continue to get an improvement in retrieval effectiveness with each additional expert added. What may be questioned is the magnitude of the increase, but our primary concern in this experiment was the demonstration that there was never a *decrease* in retrieval performance with the addition of more retrieval experts. Our second experiment is a further investigation into read depth, again using read levels of 500, 1,000, 3,000, 5,000 and 10,000. This experiment is utilising an entire retrieval run, rather than a single topic, the results

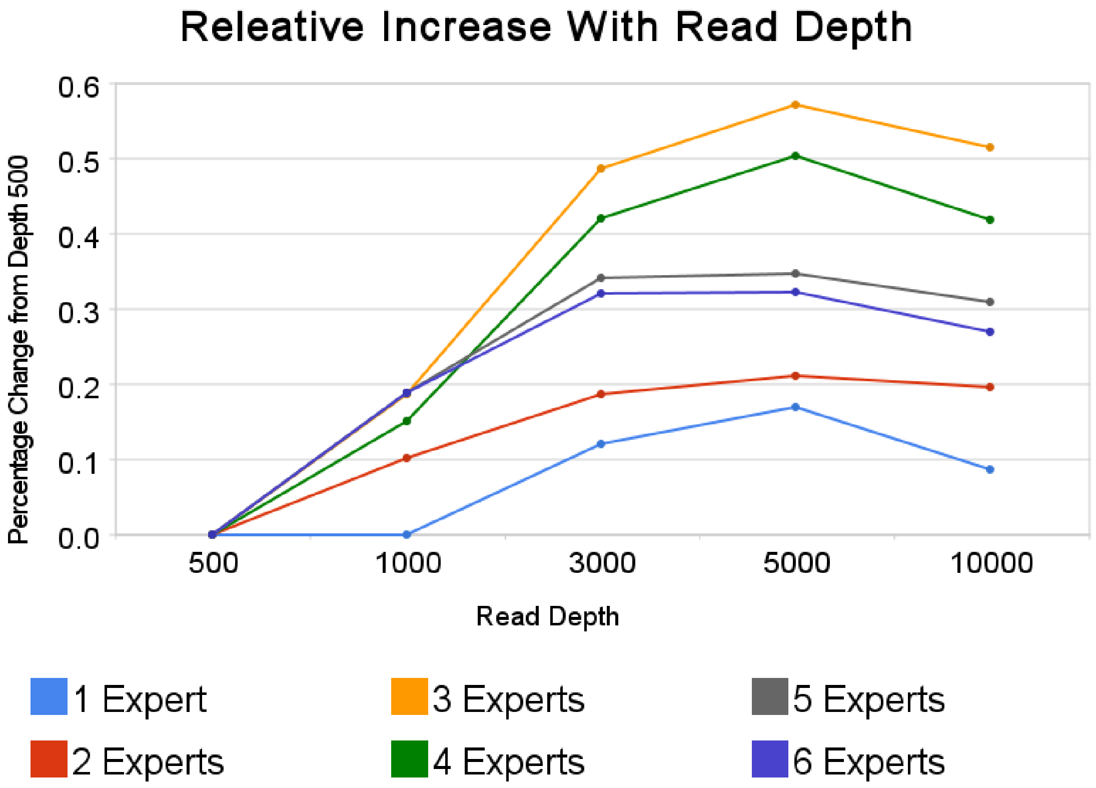


Figure 4.9: Topic 0165 from TRECVID 2005, Depth and Expert variance

are presented in Figure 4.10.

The results presented generally confirm that across evaluations, increasing the read depth for individual experts up to a level of 3,000/5,000 results in considerable performance improvements, whilst increases in depth after this point tend to saturate and level off. Certainly this relationship between read-depth, or its more conventional name recall, and precision is long established, as Cleverdon remarks there is “an inevitable inverse relationship between recall and precision” (Cleverdon et al., 1966). It is well established that as we increase recall we lower precision, although because of our optimisation we find that this behaviour instead of lowering precision, increasing recall has little positive effect on precision.

Interestingly there are some outliers, notably the TRECVID 2006 corpus, which continues to improve in performance with the increase in read-depth. TRECVID

Legend	TRECVID 2003		TRECVID 2004	
Depth	**MAP**	**Recall**	**MAP**	**Recall**
500	0.1142	0.2984	0.0919	0.2312
1000	0.1278	0.3103	0.1084	0.2312
3000	0.1437	0.3060	0.1259	0.2279
5000	0.1468	0.3259	0.1274	0.2502
10000	0.1476	0.3155	0.1312	0.2557

Legend	TRECVID 2005		TRECVID 2006	
Depth	**MAP**	**Recall**	**MAP**	**Recall**
500	0.1317	0.1656	0.0507	0.1433
1000	0.1431	0.1796	0.0581	0.1521
3000	0.1573	0.1934	0.0697	0.1608
5000	0.1595	0.2040	0.0750	0.1732
10000	0.1529	0.2139	0.0798	0.1821

Legend	TRECVID 2007		ImageCLEF 2007	
Depth	**MAP**	**Recall**	**MAP**	**Recall**
500	0.1186	0.3008	0.2063	0.4239
1000	0.1294	0.3201	0.2156	0.4398
3000	0.1457	0.3484	0.2264	0.4509
5000	0.1487	0.3639	0.2265	0.4481
10000	0.1504	0.3682	0.2253	0.4812

Figure 4.10: Depth examination, all corpora

Type	TV2003	TV2004	TV2005	TV2006	TV2007
RKF	0.2243	0.1638	0.1764	0.0985	0.1433
SubShots	0.2232	0.1774	0.1668	0.0989	0.1562

Table 4.7: Comparison of Shots vs SubShots

2006 is the lowest performing corpora we have, this continued increase likely points towards the query-images which are used as visual examples for the topics are not adequately capturing the desired information need in a form which can be exploited by the indexes.

What is of interest however is that it is a read-depth of 3,000 documents per expert, rather than 1,000 documents where this saturation begins. The level of 5,000 documents could also be considered to be this point where saturation begins, we believe it is up to the system builder to determine which of these levels to utilize in the trade-off of processing requirements versus retrieval performance gains, however in either case performance is clearly superior to 1,000 documents as the retrieval level. Whilst for the remainder of this chapter we persist with using 1,000 documents as our read-depth for extracting results from experts, particularly as this allows for greater compatibility with the text expert, the general position that extracting 1,000 documents per expert is adequate can be revised upwards.

The other variable under consideration was for digital video retrieval, and if extracting more than one keyframe per shot lead to large performance increases. We created two runs for each of the TRECVID evaluations, the first being one keyframe extracted per shot (RKF), the second being multiple keyframes extracted per shot (SubShot). Where a shot had multiple keyframes being scored we took the maximum value and used that as the final score for that shot. Our results are presented in Table 4.7.

From this data we can see that there is a difference generally favouring subshots, but that this difference is not particularly large. Of note is the performance of TRECVID 2005 which performed better with only a single keyframe extracted from the shots. For two of our evaluations, TRECVID 2003 and 2004 it is not until the third significant number that we begin to see a difference between the two. This result is both interesting and not-interesting at the same time. Effectively it is corpora dependent, in particular on the average length of a shot. We see that for news video, where the shot length is not great, there is no substantial difference between single keyframe and multiple keyframe sampling. This in effect means that for corpora of short shot-length, the selection of the middle frame as a representative image of the shot is appropriate, and no further advanced techniques are necessarily required. Conversely, the TRECVID 2007 corpus has the longest average shot length, and here we can see a marked difference between using single versus multiple keyframes. Therefore, the argument presented is that the selection of the sampling strategy for extracting frames from shots must be dependent on the average shot length. Where the average shot length is quite long, multiple or more intelligent sampling will be required. In the remainder of our experiments we will use subshots where available, but this result indicates that their impact upon performance is not as great as anticipated.

4.4 Equivalence Transformations

In this section we will examine the role of equivalence transformations and their impact on retrieval performance. To recap equivalence transformations, or more generally 'normalisation', can be defined into two broad classes, score and rank based transformations. The objective of any of the normalisation approaches examined is to perform transformations on the set of $rs_{i,j}$ which are being combined, such that the effects of different factors like score ranges, numbers of retrieved results or score distributions does not adversely impact upon performance. Recall that our candidate result sets to be combined ($rs_{i,j}$), that each is comprised of documents m, where $document_m \mapsto (name, rank, score)$. Score based normalisation approaches alter the scores of documents to be combined, thereby allowing combination to occur utilising scores, whilst conversely rank based transformations alter the ranks such that the rank of a document is what is used for combination.

Z-Score Score based transformation which converts the score of a document into the Standard Score (McClave and Sincich, 2006) (Z-Score) within the $rs_{i,j}$ from which it came, the Standard Score being a measurement of how many standard deviations a score is from the mean score. This approach has no range restriction.

Min-Max MinMax normalisation considers the best and worst scoring documents of a given $rs_{i,j}$ and assigns these scores of 1 and 0 respectively. Scores are then normalised within the range [0:1].

Borda Borda ranked based transformation, given a set size of N the Borda transformation assigns a score of $N - x$, where x is the rank. As N is the size of the result set being transformed, it produces lower scores for result sets which do not contain many documents.

BordaMAX Extends Borda, where N becomes the value of the size of the largest result set being combined, which is then used for all result sets. This is to discount the bias against small result sets present in the standard Borda approach.

RankMM MinMax normalisation based upon ranks. Conceived as a middle ground between Borda and BordaMAX, RankMM normalises the range of ranks between [0:1].

Reciprocal Reciprocal rank is a rank transformation which is heavily biased towards the top ranked documents. The rank transformation is $\frac{1}{rank_x}$

For these experiments, documents were combined using CombSUM, CombSUM being a linear combination operator. With the exception of the Z-Score approach, all approaches combined both text and visual experts. Because the Z-Score approach does not perform any range restriction, we restricted this approach to visual expert

	Score-Based		Rank-Based			
Year	Z-Score*	Min-Max	Borda	BordaMAX	RankMM	Reciprocal
TV2003	0.0095	0.1958	0.2127	0.2232	0.2206	0.1462
TV2004	0.0040	0.1524	0.1655	0.1774	0.1673	0.0892
TV2005	0.0117	0.1554	0.1654	0.1668	0.1670	0.1069
TV2006	0.0089	0.0846	0.0962	0.0989	0.0969	0.0645
TV2007	0.0195	0.1338	0.1433	0.1562	0.1459	0.0808
IC2007	0.0093	0.3148	0.3480	0.3678	0.3491	0.2503

Table 4.8: Normalisation Results Comparison. n.b.* values for approach Z-Score are based on visual experts only.

only combination so that result set sizes would not be a factor. The results of our experiment are presented in Table 4.8 with accompanying statistical tests presented in Figure 4.11.

From these results we can infer that rank based approaches generally outperform any of the score based approaches. The Z-Score method is a distinct failure with these data sets, and is excluded from further analysis. Likewise for the rank based approaches, the Reciprocal method performs poorly, although nowhere near as bad as the Z-Score approach. The failure of the Z-Score approach is likely due to the lack of any range restriction of the scores being combined. This would have made the optimisation process problematic, as there would be no equivalence between the documents to be combined, meaning that any weight generated would be required to factor this in. In the case of the reciprocal method, its failure, whilst less severe than that of the Z-Score approach, highlights that the in-built aggressive weighting of the top ranked documents of any result set was too severe (see Chapter 3 Figure 3.4). Given that earlier we established that the ideal weights for expert combination take the form of a log-normal distribution, the optimisation process would have had to devise a weight which appropriately scaled the result sets to be combined whilst discounting against the heavily weighted top documents of each result set.

From our remaining four approaches, MinMax and the three Borda variants, we can see demonstrated that the score based approach of MinMax is outperformed by all of the Borda based approaches. To compare the three Borda variants, we

$\rho = 0.05$		TV2003				TV2004			
Legend		MM	B	BM	RMM	MM	B	BM	RMM
MinMax	(MM)	-	≡	≪	≪	-	≪	≪	≪
Borda	(B)	≡	-	≪	≡	≫	-	≪	≡
BordaMAX	(BM)	≫	≫	-	≡	≫	≫	-	≡
RankMM	(RMM)	≫	≡	≡	-	≫	≡	≡	-

$\rho = 0.05$		TV2005				TV2006			
Legend		MM	B	BM	RMM	MM	B	BM	RMM
MinMax	(MM)	-	≪	≪	≪	-	≪	≪	≪
Borda	(B)	≫	-	≡	≡	≫	-	≪	≡
BordaMAX	(BM)	≫	≡	-	≡	≫	≫	-	≡
RankMM	(RMM)	≫	≡	≡	-	≫	≡	≡	-

$\rho = 0.05$		TV2007				IC2007			
Legend		MM	B	BM	RMM	MM	B	BM	RMM
MinMax	(MM)	-	≪	≪	≪	-	≪	≪	≪
Borda	(B)	≫	-	≪	≪	≫	-	≪	≡
BordaMAX	(BM)	≫	≫	-	≫	≫	≫	-	≫
RankMM	(RMM)	≫	≫	≪	-	≫	≡	≪	-

Figure 4.11: Significance Testing, Equivalence Transformations

can observe that in terms of performance the standard Borda approach is the worst performer of the three. Given that the text expert often would not have returned the full 1000 results, the standard Borda approach would have penalised this evidence. In terms of significance, the standard Borda approach is significantly worse than the BordaMAX approach, whilst for TRECVID corpora 2003-2006 there is no significant difference between it and the RankMM approach.

The BordaMAX approach appears to generally be the best performer, however there is no statistical difference between it and the RankMM approach for TRECVID corpora 2003-2006. Indeed in the case of TRECVID 2005 RankMM claims the top result. The main interpretation of this result is that rank based approaches produce very successful results, but should not penalise result sets being combined which are of smaller size.

Whilst not a bad performer in itself, across all collections the MinMax approach is significantly worse than all of the Borda based approaches (with the exception of the standard Borda approach for TRECVID 2003, however in raw performance terms standard Borda is still superior). We note that the combination of results in this experiment was utilising CombSUM. The question of why the score based approach is less successful than the rank based approaches we believe is due to the non-linear nature of scores and the application of linear weighting.

Given a result set $rs_{i,j}$, the sorted scores of this set are unlikely to exhibit a linear progression, and as such introduce a large degree of variability between the relative performance of individual documents. That is, given any two adjacent ranked documents in $rs_{i,j}$ the difference between the scores will vary considerably between pairs, whilst for ranks the difference between pairs will always be constant. Once a linear weight is then introduced, the impact of this weight will vary dependant upon the value of the score to which it is being applied, whereas for rank approaches the impact of the weight is constant and predictable. This effect has been previously identified by Lee (1997), which he termed the "independent weighting effect" which effectively introduces a second weight for combination. Upon reflection we can see that this observation is equally relevant for both the Z-Score and reciprocal approaches. In the case of the Z-Score approach, the independent weighting effect is present in the variation of the range which scores can take, whilst for the reciprocal approach it is evidenced in the aggressive weighting of highly ranked documents.

The question of the use of score or rank for data fusion is one which is not always fully considered by researchers. Many papers suggest that the use score's are better than rank as they provide more information, such as the distribution of values or other variables (Renda and Straccia, 2003)(McDonald and Smeaton, 2005)(Croft, 2000), whilst many others would provide no justification at all, as described by Lee (1997). One reason for this is that much of the work in data fusion, particularly in the text domain, has been the non-weighted combination of similarly performing retrieval experts, where the scores (similarity values) once normalised have a degree of cross-comparability, as the experts from which they came are utilising similar retrieval techniques on the same types of index (Beitzel et al., 2004). CBMIR alternatively combines experts of wildly varying performance from completely different indexing representations, and as such perhaps too much value is inferred onto the benefit of using scores from research experience found in text IR applications. Indeed as authors such as Robertson (2007) and Dwork et al. (2001) note that the score in many cases is just an artefact which is utilised for generating the ordering of a ranked list, the value of a score itself beyond this function is meaningless.

In Lee's seminal work on data fusion he hypothesized that rank combination rather than score combination should perform better, given that scores are impacted by the "independent weighting effect". However experimentally he found that this is not the case (Lee, 1997), his work using normalised scores provides the better results. Croft (2000) provides an interpretation of Lee's result:

> "This can be interpreted as evidence that the normalised score is usually a better estimator for the probability of relevance than the rank. Using the ranks is a more drastic form of smoothing that appears to increase error except when the systems being combined have very different scoring characteristics" (Croft, 2000).

This interpretation fits exactly to the characteristics of CBMIR, where as previously established we have very noisy sources of evidence being combined which vary wildly in performance. In the final section of this chapter we will revisit Lee's experiments and offer explanations as to why his initial hypothesis may have actually been correct.

4.5 Combination Operators

In this section we will examine the difference in performance between the two most common combination operators for data fusion, CombSUM and CombMNZ (Fox and Shaw, 1994). To recap, for both of these operators we are examining their weighted form, that is each individual component is first weighted, then combined. These

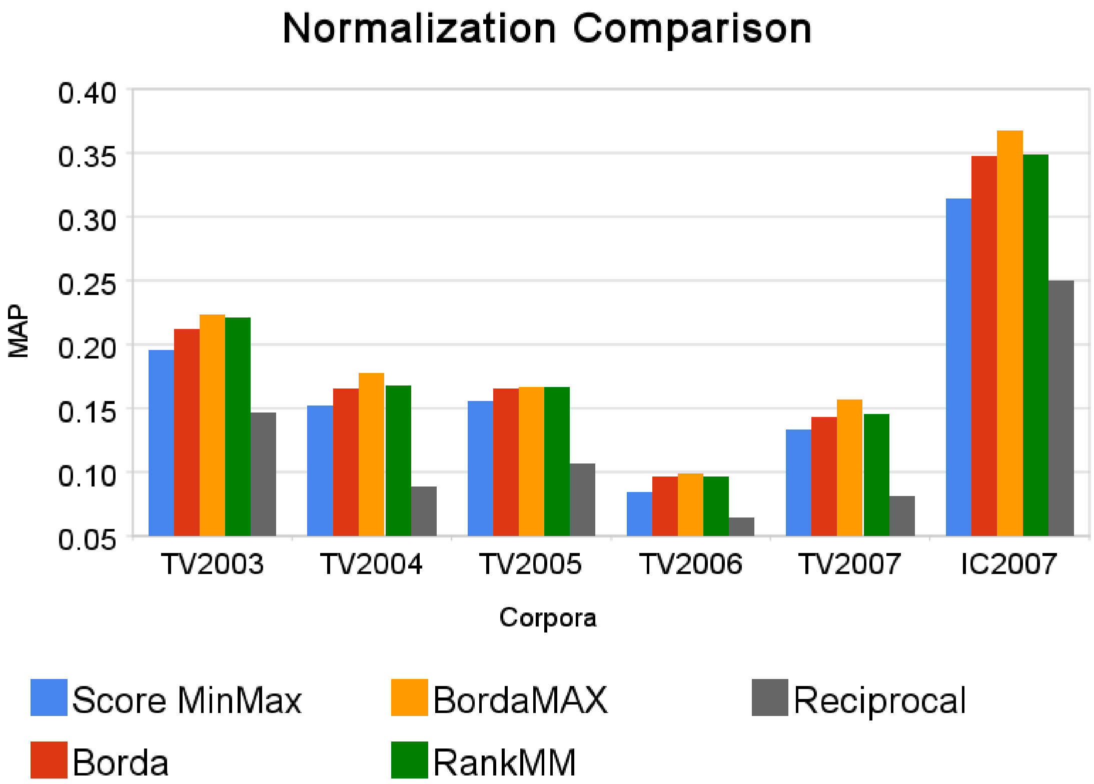


Figure 4.12: Comparison of Normalisation approaches, n.b. Z-Score approach not shown.

	CombMNZ		CombSUM	
Year	Score	Borda	Score	Borda
TV2003	0.1720	0.1458	0.1958	0.2127
TV2004	0.1321	0.1258	0.1524	0.1655
TV2005	0.1329	0.1230	0.1554	0.1654
TV2006	0.0770	0.0701	0.0846	0.0962
TV2007	0.1157	0.1131	0.1388	0.1433
IC2007	0.2991	0.2861	0.3148	0.3480

Table 4.9: CombSUM vs. CombMNZ for score & rank based approaches.

operators utilise the normalised scores of result sets (see previous section), therefore they work with either score or rank based approaches. CombSUM is the summation of the weighted normalised scores to be combined, whilst CombMNZ extends this by adding an additional variable, the number of times a document appears in the result sets and multiplies this value against the CombSUM value. In behavioural terms, CombSUM can be thought of as a linear operation, whilst CombMNZ produces a non-linear response. Full details on these operators are provided in the previous Chapter (see section 3.6).

For this experiment, we will investigate the impact of CombSUM and CombMNZ on both score and rank based normalisation approaches. To test the score based approach we will utilise the MinMax score normalisation technique. For rank we will utilise the standard Borda approach. The experts used for combination involve both text and visual experts and result sets $rs_{i,j}$ are truncated to 1,000 results. The results of this experiment are presented in Table 4.9, visualized in Figure 4.14 with significance data presented in Figure 4.13.

There is little ambiguity in the presented results as we can quite clearly observe a constant ordering across all corpora with regards to performance. CombSUM utilising Borda normalisation is the most effective combination operator, whilst CombMNZ using ranks is consistently the worst performer.

However there is one curious artefact, which may explain the preponderance of data fusion literature which advocates the use of CombMNZ. From our results we ob-

$\rho = 0.05$		TV2003				TV2004			
Legend		SCZ	BCZ	SCS	BCS	SCZ	BCZ	SCS	BCS
Score MNZ	(SCZ)	-	$\gg$	$\ll$	$\ll$	-	$\equiv$	$\ll$	$\ll$
Borda MNZ	(BCZ)	$\ll$	-	$\ll$	$\ll$	$\equiv$	-	$\ll$	$\ll$
Score SUM	(SCS)	$\gg$	$\gg$	-	$\equiv$	$\gg$	$\gg$	-	$\ll$
Borda SUM	(BCS)	$\gg$	$\gg$	$\equiv$	-	$\gg$	$\gg$	$\gg$	-

$\rho = 0.05$		TV2005				TV2006			
Legend		SCZ	BCZ	SCS	BCS	SCZ	BCZ	SCS	BCS
Score MNZ	(SCZ)	-	$\gg$	$\ll$	$\ll$	-	$\gg$	$\ll$	$\ll$
Borda MNZ	(BCZ)	$\ll$	-	$\ll$	$\ll$	$\ll$	-	$\ll$	$\ll$
Score SUM	(SCS)	$\gg$	$\gg$	-	$\ll$	$\gg$	$\gg$	-	$\ll$
Borda SUM	(BCS)	$\gg$	$\gg$	$\gg$	-	$\gg$	$\gg$	$\gg$	-

$\rho = 0.05$		TV2007				IC2007			
Legend		SCZ	BCZ	SCS	BCS	SCZ	BCZ	SCS	BCS
Score MNZ	(SCZ)	-	$\equiv$	$\ll$	$\ll$	-	$\equiv$	$\ll$	$\ll$
Borda MNZ	(BCZ)	$\equiv$	-	$\ll$	$\ll$	$\equiv$	-	$\ll$	$\ll$
Score SUM	(SCS)	$\gg$	$\gg$	-	$\ll$	$\gg$	$\gg$	-	$\ll$
Borda SUM	(BCS)	$\gg$	$\gg$	$\gg$	-	$\gg$	$\gg$	$\gg$	-

Figure 4.13: Significance Tests for CombSUM vs. CombMNZ approaches

serve that when scores (MinMax) rather than ranks are used, CombSUM achieves greater performance than CombMNZ once fully optimised. Whilst in raw MAP terms this effect always holds, in terms of significance tests only TRECVID collections 2003, 2005 and 2006 demonstrate a significant difference between CombSUM and CombMNZ when using scores. But when CombMNZ is considered in isolation, score based CombMNZ consistently outperforms rank-based CombMNZ, reversing the observation found with CombSUM. This result is of interest particularly when coupled with our previous observations about the popularity of using scores for data fusion in the research literature (Lee, 1997; Croft, 2000).

Given that the use of scores for data fusion is prevalent, with the exception of metasearch applications where scores are assumed not to be available, CombMNZ is more effective when using score based normalisation rather than using rank based normalisation, perhaps indicating why it has found such popularity in the data fusion community. Despite this however, if we examine the performance of score based

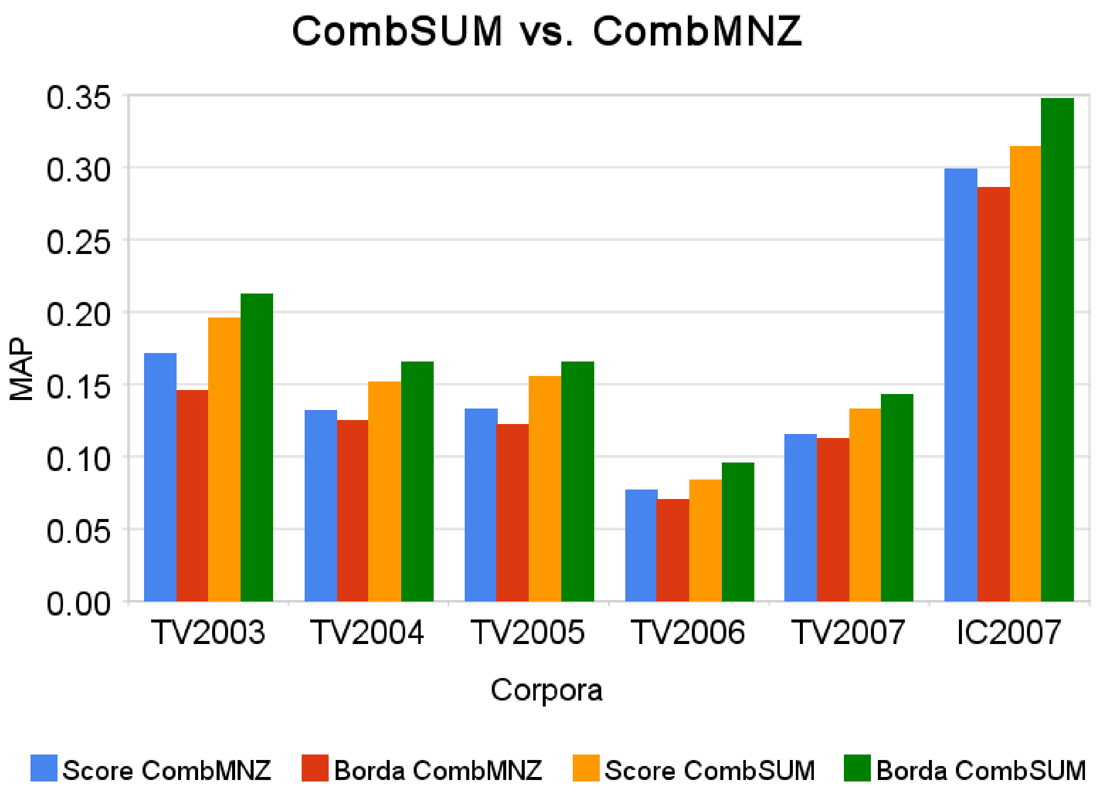


Figure 4.14: Performance Comparison: CombSUM vs CombMNZ, score and rank based approaches.

CombMNZ versus score based CombSUM, we find that CombSUM is consistently better. This level of performance is again improved if rank normalisation rather than score normalisation is used in conjunction with CombSUM. Furthermore there is always a significant difference in performance between score based CombMNZ and score based CombSUM, with weighted CombSUM clearly outperforming CombMNZ.

What this means is that if a researcher examining data fusion takes the position *a priori* that CombMNZ will be superior to CombSUM, based upon published research in data fusion from the text IR domain, they will maximise its potential by making use of score rather than rank normalisation techniques. This in turn leads to a situation where CombMNZ and score normalisation becomes the default case for conducting data fusion experiments. We have empirically demonstrated, certainly for the case for CBMIR, that to maximise performance for weighted linear combination that CombMNZ and score based normalisation are not appropriate default positions for maximising performance. Furthermore as indicated previously, we will revisit Lee's early experiments and demonstrate that the case for CombMNZ and score normalisation may be overstated for text IR as well.

4.6 Combination Levels

One of the major implicit variables in CBMIR is that of combination levels, that is, the aggregation of results at different parts of a query in order to form a single response. In this section we will be examining three classes of aggregation previously detailed in Section 3.7 of Chapter 3. The three levels to be examined are 'query level', 'expert level' and 'direct level'. In our CBMIR system for any given query we will have available $|E|$ experts and $|Q|$ query components. For 'expert level' combination, we take each query component and issue it against an expert, combining the results uniformly for that given expert to create a single result set which represents the results for that expert (rs_i). Then each of these 'expert sets' can then be weighted and combined to form a response to the query. Alternatively for 'query level' combination, for every query component we query it against every expert and uniformly combine the results, creating for each query component a single result set (rs_j). Each of these aggregated query component result sets can then be weighted and combined to form a response to the query. Direct level combination involves no combinatorial hierarchy, it is the direct weighting of every unique result set $rs_{i,j}$. A complete discussion of these levels is found in the previous chapter in Section 3.7.

To simplify the task we restrict our investigation to visual only experts, so as to avoid issue of result set size impacts. However, for the majority of our experiments we have run both text and visual, and visual only versions and whilst there are differences in performance, the patterns of performance remain constant, i.e. observations from text and visual experiments are consistent with observations from visual experts only experiments. We use MinMax score normalisation for our investigation, with CombSUM used for combination and linear weighting.

For each experiment we include the minimum and maximum achieved for that corpus. The minimum is a 'Uniform' run, where all pairs $\langle Expert_i, Query_j \rangle$ are equally weighted, demonstrating the performance achieved if no weighting scheme at all is employed. The maximum is the fully optimised result, demonstrating the best performance that can be achieved. These two figures provide a lower and upper bound for data fusion performance comparisons, allowing us to make decisions using absolute observations with regard to the bounds, rather than relative observations by comparing only to existing data fusion approaches. As a reference we also include two additional runs featured earlier in this chapter, the runs 1σ and 1σ-U. These runs are those which *only* make use of highly weighted pairs $\langle Expert_i, Query_j \rangle$, the run 1σ using the ideal weights that were assigned to these pairs, and the run 1σ-U which uses only these pairs and assigns them a uniform weight.

Our results are presented firstly in terms of MAP in Figure 4.15 and visualized in Figure 4.17. For the MAP results, next to each run in parenthesis we display how close in terms of performance the run came to matching the fully optimised score. Significance test for the four runs examined are given in Figure 4.16.

One of the major purposes in conducting this experiment is that many existing approaches for data fusion would impose some form of a hierarchy when implementing a weighting scheme. This is understandable as the task of weighted data fusion is difficult, and levels of aggregation allow for a degree of generalisation to occur in setting weights for a retrieval scenario. What has not been explored however to our knowledge is any form of cap that the imposition of a hierarchy can place upon performance. From the retrieval figures presented here we can clearly see that with respect to optimal weighting that quite a significant upper bound is place on performance.

On average, both the query and expert level approaches achieve only 58% of the theoretical maximum performance achievable, keeping in mind that this is after both of these approaches has been fully optimised, meaning that any data fusion

Legend	TRECVID 2003		TRECVID 2004	
Run	**MAP**	**Recall**	**MAP**	**Recall**
Uniform All	0.0593	0.2375	0.0288	0.1440
Expert Level	0.0752 (61%)	0.2653	0.0519 (47%)	0.1684
Query Level	0.0776 (63%)	0.2729	0.0543 (50%)	0.2018
1σ Uniform	0.0966 (79%)	0.2786	0.0738 (68%)	0.2268
1σ	0.0989 (80%)	0.2805	0.0770 (71%)	0.2246
All Optimised	0.1224	0.3027	0.1084	0.2318

Legend	TRECVID 2005		TRECVID 2006	
Run	**MAP**	**Recall**	**MAP**	**Recall**
Uniform All	0.0646	0.1140	0.0164	0.0926
Expert Level	0.0827 (59%)	0.1344	0.0299 (53%)	0.1138
Query Level	0.0850 (60%)	0.1456	0.0262 (47%)	0.1150
1σ Uniform	0.1037 (74%)	0.1484	0.0460 (82%)	0.1332
1σ	0.1108 (79%)	0.1574	0.0496 (88%)	0.1393
All Optimised	0.1407	0.1725	0.0563	0.1493

Legend	TRECVID 2007		ImageCLEF 2007	
Run	**MAP**	**Recall**	**MAP**	**Recall**
Uniform All	0.0422	0.2007	0.1283	0.4095
Expert Level	0.0655 (56%)	0.2500	0.1648 (76%)	0.4133
Query Level	0.0700 (60%)	0.2614	0.1544 (71%)	0.4148
1σ Uniform	0.0680 (58%)	0.2840	0.1404 (65%)	0.3809
1σ	0.0772 (66%)	0.2615	0.1439 (68%)	0.3814
All Optimised	0.1175	0.3121	0.2156	0.4379

Figure 4.15: Combination Levels Performance Comparison

approaches which implement one of these combinatorial hierarchies places a significant ceiling on the performance that can be attained. This contrasts to the average performance of our target weighted runs, where rather than a combination level we used instead specific pairs $\langle Expert_i, Query_j \rangle$ from the matrix $\mathbf{RS}$. On average this targeted approach achieves 73% of the theoretical maximum achievable. When contrasted against the maximum performance attainable when using a combination level, this result emphasises our findings earlier in this chapter, that targeted weighting of specific pairs $\langle Expert_i, Query_j \rangle$ can produce considerable performance gains. Of interest is that with the exception of the Clef2007 corpora, that there is no significant difference between the query or expert combination levels.

The Clef2007 corpus is something of an outlier in this set of experiments. Throughout the experiments in this chapter we have typically found broadly similar behaviours among each of our test corpora. In this case however we notice a significant deviation, which to a lesser extent is also present in the TRECVID 2007 corpus. In the Clef2007 corpora, the expert level performs clearly the best with performance significantly greater than all other approaches. The cause for this is unclear, one artefact of note is that this collection has the highest density of query topic images to the collection, with one query image for every 6667 collection images per topic. This indicates that recall played a more prominent role in the corpus, and that the selection of highly-weighted pairs may have been too restrictive to provide adequate topic coverage.

Our fundamental conclusions from this section however remain that with regards to combination level, for the most part, there is no significant difference between either the query or expert combination levels. What has been established is the relatively low cap that these levels place on the maximum performance that can be achieved employing either of those two combination levels. That is, even if the ideal set of weights was employed in a query or expert combination level weighting scheme, that we are on average likely to only obtain 58% of the performance of the ideal direct level combination's performance. This result reinforces our earlier

$\rho = 0.05$		TV2003				TV2004			
Legend		E	Q	1σ-U	1σ-O	E	Q	1σ-U	1σ-O
Expert Level	(E)	-	$\equiv$	$\ll$	$\ll$	-	$\equiv$	$\ll$	$\ll$
Query Level	(Q)	$\equiv$	-	$\ll$	$\ll$	$\equiv$	-	$\ll$	$\ll$
1σ Uniform	(1σ-U)	$\gg$	$\gg$	-	$\equiv$	$\gg$	$\gg$	-	$\equiv$
1σ Optimised	(1σ-O)	$\gg$	$\gg$	$\equiv$	-	$\gg$	$\gg$	$\equiv$	-

$\rho = 0.05$		TV2005				TV2006			
Legend		E	Q	1σ-U	1σ-O	E	Q	1σ-U	1σ-O
Expert Level	(E)	-	$\equiv$	$\ll$	$\ll$	-	$\equiv$	$\ll$	$\ll$
Query Level	(Q)	$\equiv$	-	$\ll$	$\ll$	$\equiv$	-	$\ll$	$\ll$
1σ Uniform	(1σ-U)	$\gg$	$\gg$	-	$\ll$	$\gg$	$\gg$	-	$\ll$
1σ Optimised	(1σ-O)	$\gg$	$\gg$	$\gg$	-	$\gg$	$\gg$	$\gg$	-

$\rho = 0.05$		TV2007				IC2007			
Legend		E	Q	1σ-U	1σ-O	E	Q	1σ-U	1σ-O
Expert Level	(E)	-	$\equiv$	$\equiv$	$\equiv$	-	$\gg$	$\gg$	$\gg$
Query Level	(Q)	$\equiv$	-	$\equiv$	$\equiv$	$\ll$	-	$\gg$	$\equiv$
1σ Uniform	(1σ-U)	$\equiv$	$\equiv$	-	$\ll$	$\ll$	$\ll$	-	$\ll$
1σ Optimised	(1σ-O)	$\equiv$	$\equiv$	$\gg$	-	$\ll$	$\equiv$	$\gg$	-

Figure 4.16: Significance Tests for Combination Level variations

observations about the distribution of highly weighted pairs and the importance of attempting to specifically upweight specific pairs $\langle Expert_i, Query_j \rangle$ in order to obtain high performance.

4.7 Lee's TREC-3 Data Fusion Experiments

One of the major papers in data fusion is that by Lee, who performed a series of data fusion experiments investigating the role of various combination operators (Lee, 1997). This paper is highly cited (as of May 2009, 311 citations according to Google scholar), and serves as the primary justification for utilising techniques such as MinMax score normalisation and the use of CombMNZ. It was this paper that formulated the hypothesis as to why data fusion works, namely that:

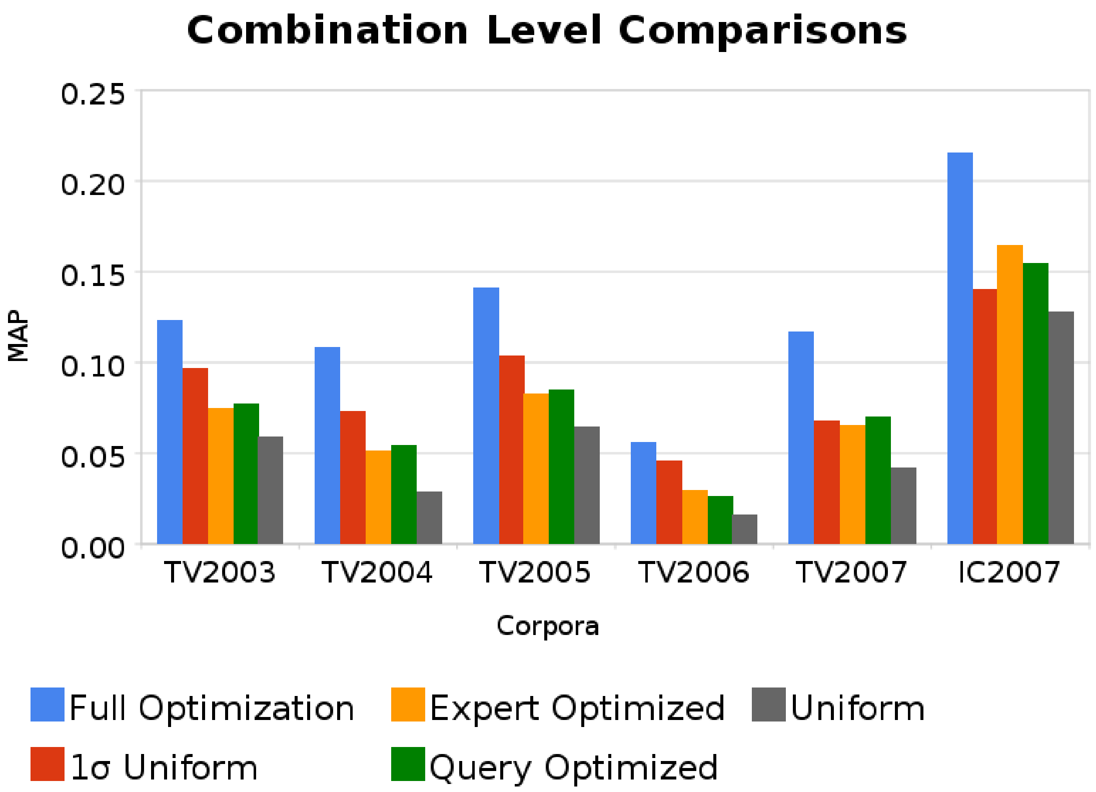


Figure 4.17: Combination Levels Performance

> different runs might retrieve similar sets of relevant documents but retrieve different sets of non-relevant documents (Lee, 1997).

This hypothesis was arrived at through the examination of prior work, notably Belkin et al. (1993) who observed that combinations of multiple query formulations of the same information need led to performance improvements, Turtle and Croft (1991) who noted overlaps of relevant documents returned by between retrieval systems, and finally Saracevic and Kantor (1988) who found that the more runs a document is retrieved by the higher it should be ranked. All this informed the creation of the data fusion hypothesis and the establishment of the overlap metrics which we previously explored.

Lee examined the earlier work of Fox and Shaw (1994) where the majority of combination operators for data fusion were originally defined. Fox and Shaw found that of the approaches they explored, CombSUM and CombMNZ performed well, with CombSUM performing the best. This contention was one which Lee felt was incorrect, as he thought that documents which appear more often in ranked lists should be promoted higher up the final ranking, that is CombMNZ should work better than CombSUM.

To investigate this, Lee ran a series of experiments on the combination of multiple text retrieval runs from the TREC-3 ad-hoc retrieval task (Harman, 1993). For reference, the runs which Lee combined were (TREC identifiers): *westp1*, *pircs1*, *vtc5s2*, *brkly6*, *eth001* and *nyuir1*. In his experiments, retrieval runs were first normalised using MinMax then linearly combined using either CombSUM or CombMNZ. There was no weighting used in the combination.

Lee's major conclusions from this study were that CombMNZ provided better retrieval effectiveness than the other combination methods (e.g. CombSUM), and that score normalisation approaches typically worked better than rank aggregation, except in cases where the retrieval runs being combined have different distributions of normalised scores.

	CombMNZ		CombSUM	
Experiment	Score	Borda	Score	Borda
Lee Orig. (no weights)	0.3991	0.3915	0.3972	0.3934
Weighted data fusion	0.4567	0.4461	0.4620	0.4621

Table 4.10: TREC-3 Six System Data Fusion Results

Lee's observations appear to run counter to what we have established empirically in this chapter. Given that Lee's work informs a large body of data fusion research, it is appropriate that we revisit the experiments to determine why there are differences between our observation and that of Lee's.

We devise two sets of experiments for investigation, the first being a repeat of Lee's six system combination of TREC-3 data, whilst the second is the combination of the six systems, but utilising our weighted optimisation techniques so as to determine what the best combination of these runs would produce. For each of these two experiments we test the effect of CombSUM and CombMNZ, using both Borda rank and MinMax score normalisation. As the majority of the topics for all systems returned 1,000 results, standard Borda count is utilised as it will not have an adverse impact upon performance. Results from these experiments are presented in Table 4.10, visualized in Figure 4.18 and significance tests are reported in 4.19.

We note that our reproduction of Lee's results produced the same MAP scores as reported in his work (Lee, 1997). For the unweighted combination, the highest performer was indeed score based CombMNZ, followed by score based CombSUM, then the two Borda approaches. However an examination of the statistical significance tests for this runs paints a slightly different picture. The only certainty from the unweighted experiment was that the run Borda CombMNZ was significantly the worst run of the four, with each of the other three runs out-performing it. This is the same as our experimental findings where Borda CombMNZ was our worst performer. The runs score CombSUM and score CombMNZ exhibit no statistically significant difference, indicating that the reported difference between the two is likely due to chance.

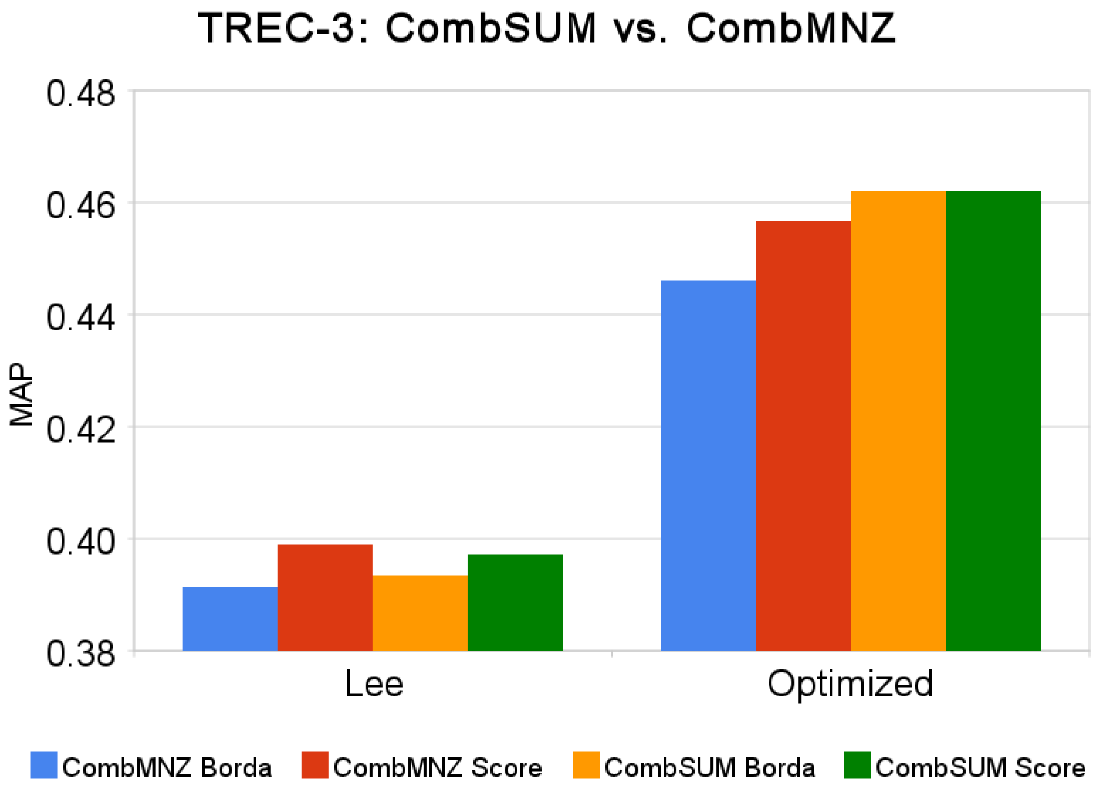


Figure 4.18: TREC-3 Data Fusion Performance

$\rho = 0.05$		TREC 3 - No Weights				TREC 3 - Weighted			
Legend		SCZ	BCZ	SCS	BCS	SCZ	BCZ	SCS	BCS
Score MNZ	(SCZ)	-	≫	≡	≫	-	≫	≪	≪
Borda MNZ	(BNZ)	≪	-	≪	≪	≪	-	≪	≪
Score SUM	(SCS)	≡	≫	-	≡	≫	≫	-	≡
Borda SUM	(BCS)	≡	≫	≡	-	≫	≫	≡	-

Figure 4.19: Significance Tests for TREC-3 experiments

The optimised runs present an entirely different set of results which contradicts the conclusions of Lee regarding the superiority of CombMNZ as a combination operator. Examining the statistical difference between the optimised runs, we observe that both Borda CombSUM and score CombSUM are statistically significantly different to both of the CombMNZ runs, matching the observations we make in this chapter. However between the two CombSUM runs there is no statistical difference.

A reason for the difference between our findings and that of Lee's we believe is related to Lee's investigation into the effect of rank based normalisation and its application when the distribution of scores varies between systems. Lee hypothesized in his paper that rank based normalisation should perform better than score based normalisation because of what he termed the 'independent weighting effect', which is that effectively a score acts as a weight for a document. Lee found that using ranks where scores have very different score distributions did improve retrieval effectiveness. We believe that in the case we are examining here, strong performing text IR systems which work off the same 'symbols' (i.e. indexable elements), that the scores produced are relatively similar in distribution and therefore there is little difference between rank or score based normalisation if the score distributions are similar. Conversely this also explains why rank normalisation works well in our CBMIR experiments as we would expect the underlying score distributions of each result set to be quite different.

Of more interest however is the result of CombSUM and CombMNZ, particularly given the popularity that CombMNZ holds in data fusion, whilst in our experiments performing empirically worse. Lee's motivation for demonstrating the effectiveness of CombMNZ was an extension of the work of Saracevic and Kantor (1988) where they observed that a document which appears in multiple ranked lists should be ranked higher. Lee saw CombMNZ as a mechanism which gave a boost in ranking to documents which appeared in more lists.

Thinking of CombMNZ in terms of providing a boost to documents which appear in multiple lists gives an impression of a function which is positive, that it

Doc.	E1	E2	E3	CombSUM	CombMNZ
a	0.8	0.0	0.9	1.7	3.4
b	0.6	0.3	0.6	1.5	4.5

CombSUM ranking	a > b
CombMNZ ranking	b > a

Table 4.11: CombSUM vs CombMNZ behaviour

promotes documents up a ranked list. An alternative way however when thinking of CombMNZ is negative, that in fact the function's purpose is to *penalise* documents which do not appear in *all* ranked lists. We can illustrate this with a toy example given in Table 4.11.

In this table we have two documents, 'a' and 'b', and three experts 'E1 ... E3'. Document 'a' is only found in two of our three experts, however it is highly ranked positions in both of these experts. Document 'b' on the other hand is found in all three experts, but it appears around the middle of each experts result set. The combination behaviour of CombSUM and CombMNZ in this case produces two quite different orderings. CombSUM will rank document 'a' before document 'b', whereas CombMNZ will reverse the ranking putting 'b' before 'a' as it appears in all three of our result sets. The question for system builders becomes what sort of behaviour do they want present in their retrieval system

Compounding this is the non-linear behaviour that CombMNZ produces compared to the linear behaviour of CombSUM, highlighted in the previous chapter. Figure 4.20 shows a hypothetical scenario where we have six retrieval experts (as in the TREC-3 experiment) and in each expert we find the same document which is assigned the same score by each expert. This graph demonstrates how the cumulative score of that document changes as it is found in more experts for both CombSUM and CombMNZ. The third line in this graph demonstrates the relative difference in scores between CombSUM and CombMNZ.

From this graph we can clearly see that CombSUM is a linear function whilst CombMNZ is not. Compounding this, is that in relative terms, a document which

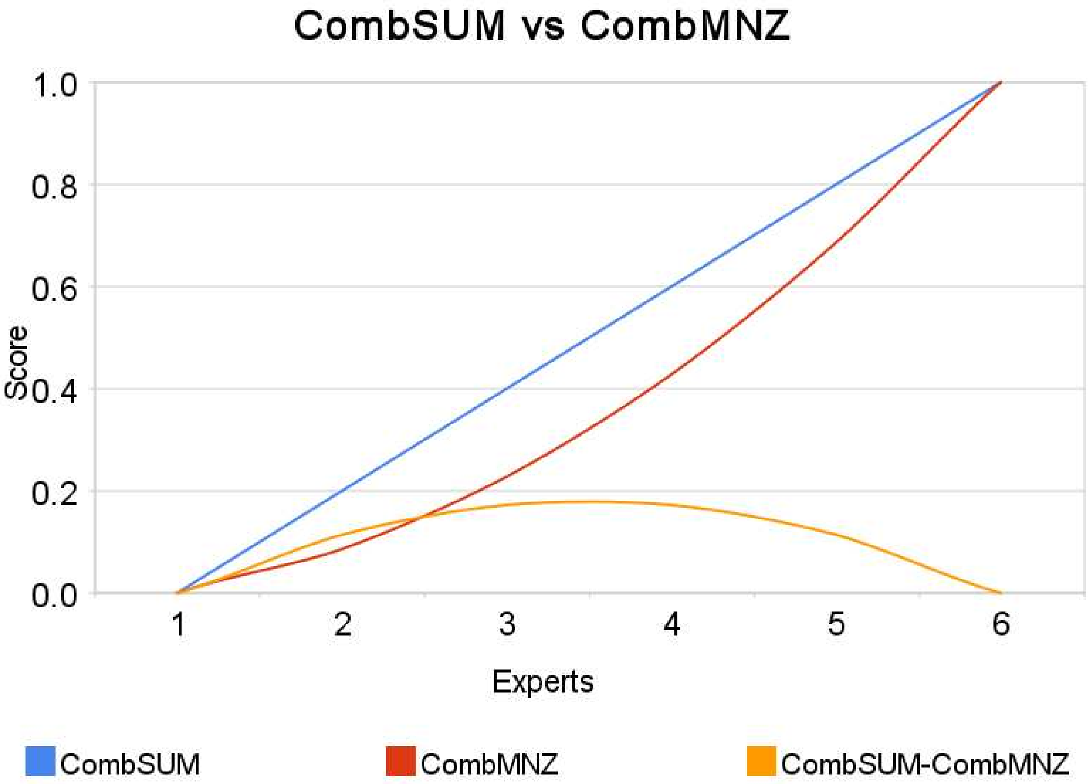


Figure 4.20: CombSUM vs CombMNZ differences

is found in either 3 or 4 experts in CombMNZ receives a larger score penalty than a document which is found in either 2 or 5 experts. This is of crucial importance, because in practice, documents will rarely be assigned the same score by a retrieval expert, meaning that once this non-linerality is taken into account the ranking becomes much more unpredictable for CombMNZ. For CombSUM as the behaviour is linear, the response to the combination can be considered more stable, that is as a document is found in more ranked lists its score will increase linearly. We believe that the CombSUM behaviour is more desirable, particularly considering that once any large number of result sets are to be combined through data fusion, that the probability of a document being found in every, or a majority of result sets, must decrease. This has been empirically justified in the results we have presented in this chapter for CombSUM and CombMNZ.

We are not the first to go back and examine various aspects of Lee's work, notably authors such as Beitzel et al. (2004) have re-examined the data fusion hypothesis of the overlaps between relevant and non-relevant documents being the key driver for retrieval performance. Beitzel *et al.* conclude that when fusing different retrieval *algorithms* within the same system, that no benefit is derived from data fusion, as relevant documents are already highly ranked. Therefore the promotion of documents up a final ranked lists is typically from overlapping non-relevant documents. Our earlier work disagrees with this sentiment, in that earlier in this chapter we found that the ratio of $R_{overlap}/NR_{overlap}$ correlated highly with average precision. Our work is not cross-comparable however, as the work of Beitzel *et al.* is dealing with high quality retrieval algorithms within the same system, whereas we are examining the combination of multiple disparate, noisy, retrieval sources.

The use of weighting in data fusion for text IR is not widespread, particularly as many text retrieval algorithms can produce comparable performance which is not subject to the wide variation in performance seen in CBMIR. In these cases the application of CombMNZ can be seen to be used as a surrogate for linear result set weighting. As we have established, CBMIR applications utilise sources of evidence which can vary wildly in performance, which requires that weighting is employed in order to maximise performance. Given that, the use of rank normalisation and CombSUM appears to be the most appropriate methods for combining ranked lists together. Furthermore we have demonstrated that for the case of text IR, using ranks and CombSUM produces equally the best weighted combination performance, indicating that if text based data fusion approaches were to adopt these operators rather than the more common CombMNZ score MinMax combination, performance improvements can be achieved.

4.8 Conclusions

In this chapter we have conducted an extensive empirical investigation into weighted data fusion and its impact on CBMIR retrieval performance. To achieve our observations we employed a non-conventional experimental model, which directly optimised retrieval performance on test corpora which were being examined. The major benefit of this approach was that for every subsequent experiment that was run within this framework, the ideal set of linear weights for data fusion was always used. This in effect held constant the effect of weighting on retrieval performance and allowed us to make robust observations as to the impact of different factors which effect the data fusion process.

The following observations were made in this chapter (summarised in Table 4.12):

- The ideal weighting form for data fusion is a highly positively skewed distribution, where specific $rs_{i,j}$ in a topic are highly weighted, whilst the remaining results receive the remainder of the weight. We found that approximately 10%-20% of the result sets being combined in a query attracted 60%-80% of the weight when ideally weighted.

- The 'read-depth' whilst by convention is set at 1,000 documents, is for the case of CBMIR too small. Given that very noisy form of evidence are being combined, increasing the 'read-depth' to 3,000 documents produces demonstrable improvements in recall and precision.

- Normalisation plays a crucial component in putting evidence into a form which allows it to be easily combined. For score based normalisation we found that MinMax normalisation produces the best results. For rank based normalisation we found that our BordaMAX or RankMM approaches produced the best performance. Overall we found that when ideal weighting is used, rank based normalisation out-performed score based normalisation.

- We examined the performance of the two most common combination operators, CombSUM and CombMNZ. Whilst CombMNZ is the most popular form of combination operator in text retrieval data fusion literature, we found that CombSUM clearly outperformed CombMNZ.

- Combination levels are often employed in data fusion approaches to make the task of combining evidence from sources more manageable. We found that combining at the 'expert' or 'query' level produced no difference in retrieval performance. Combining at the 'direct' level though produced a very large performance gain over combining results at any arbitrary level, greater than what was anticipated. Typically imposing a combination level restricted retrieval performance to between 50%-75% of what could be achieved with direct combination. This level of performance is better when we consider the first point of this list, that specific pairs correctly weighted drive retrieval performance. The direct level of combination is the only level of combination which allows for the specific weighting of pairs, and therefore is the most effective level of combination.

- We re-examined the early data fusion experiments of Lee, and found that contrary to reported results, for text retrieval when linearly weighted, CombSUM outperforms CombMNZ. Furthermore we found that despite the reporting of

Summary of Major Findings	
Ideal Weight Distribution	Highly Positively skewed Log-Normal.
Read Depth	Between 3,000-5,000 documents.
Keyframe Sampling	Dependent upon average shot length.
Equivalence Transformations	
Score based	MinMax
Rank based	BordaMAX or RankMM
Overall	Rank superior with ideal weights.
Combination Operators	CombSUM outperforms CombMNZ.
Combination Levels	Direct-level far superior to alternatives.
High Impact Queries	Approx. 10% of query-components provide 80% of performance.

Table 4.12: Summary of Major Findings

CombMNZ performing better than CombSUM when no weights are utilised, that in fact there was no significant difference between the two approaches.

This chapter has explicitly examined many data fusion variables to a very fine level of detail and discovered several properties of data fusion which were masked by sub-optimal weighting being employed. Notable was the assumptions that CombMNZ is the ideal form of evidence combination. From these observations, we find that many of the earlier examined algorithms for data fusion do not have properties which would allow for the full exploitation of the ideal data fusion form which we have observed in this chapter. Notably, none of the examined data fusion algorithms allows for the direct weighting of individual pairs $\langle Expert_i, Query_j \rangle$ at query time, with most data fusion approaches employing a combination level approach which aggregates results at the 'expert' level. In the next chapter we will review current approaches to weighted data fusion and determine if they leverage the findings we have made.

Chapter 5

Existing Approaches for Weighted Data Fusion

In this chapter we will expand upon the initial data fusion concepts and operators introduced in Chapter 3. Specifically in this chapter we will examine approaches which not only combine the outputs of various retrieval experts, but also allow for their weighted combination. As demonstrated in Chapter's 3 and 4, weighting is crucial to achieving optimal performance in the context of Content-Based Multimedia Information Retrieval (CBMIR), where individual queries and experts may perform poorly, but their successful weighted combination achieves far greater performance than the sum of their parts. We begin with a revision of past data fusion research and implementation mostly within the text information retrieval domain. This is followed by a detailed examination of combination approaches that either explicitly define or can be modified to allow for weighting of the output of specific ranked lists. Finally we finish the chapter with a brief examination of other approaches within information retrieval which utilize the combination of data but are outside the scope of this thesis. A high-level description of multimedia retrieval and the more general problems facing it is detailed in Chapter 2.

5.1 Data Fusion

To recap our definitions introduced in Chapter 3, when we refer to *Data Fusion* we are conducting “the combination of evidence from differing systems” (Belkin et al., 1995) with the aim of maximizing retrieval performance. This is a distinct from the *Collection Fusion* problem, which Voorhees *et al.* defines as the combination of “retrieval runs on separate, autonomous document collections that must be merged to produce a single, effective result” (Voorhees et al., 1995). Fusion is an overloaded term, within the multimedia processing community it can also refer to *Information Fusion* which includes activities such as the combination of various modalities such as the information from a visible light camera and an Infra-Red camera into a single signal (Kludas et al., 2008).

As further revision, we reintroduce the variables we used for describing data fusion within the context of a CBMIR system. In a CBMIR system we have available a set of multi-modal Retrieval Experts E where there are $1 \leqslant i \leqslant |E|$. The CBMIR system can process a multi-example multi-modal query Q, which is composed of multiple components j, where $1 \leqslant j \leqslant |Q|$. Each unique pairing of an expert and query component $\langle expert_i, query_j \rangle$, produces an ordered set of documents $R = \{document_1 \ldots document_m\}$, and every document has associated with it a name, rank position and real valued score. Again we note the use of the term ‘document’ to refer to the retrieval unit being used within a multimedia corpus, where a document may be any of a speech transcript, segment of video, an image etc. The processing of the query Q against the expert set E produces the sparse matrix of results **RS** where each element in the matrix $rs_{i,j}$ corresponds to the result set generated by pair $\langle expert_i, query_j \rangle$. Finally, in order to combine the elements of **RS** into a single result list, we need a corresponding matrix of weights **RC**, such that each individual element of the matrix $rc_{i,j}$ contains the weight which will be applied to the result set $rs_{i,j}$. Therefore the final result of a multi-part query is the application of the weights contained within **RC** to the result sets **RS** which are then linearly interpolated to

Retrieval Experts	$E = \{expert_1 \ldots expert_i\}$
Multi-Modal Query	$Q = \{query_1 \ldots query_j\}$
Expert-Query pair	$\langle expert_i, query_j \rangle$
Ranked Result	$R = \{document_1 \ldots document_m\}$
Document	$document_m \mapsto (name, rank, score)$
Result Set Matrix	$\mathbf{RS} = [rs_{i,j}]_{\lvert E\rvert \times \lvert Q\rvert}$
Result Coefficient Matrix	$\mathbf{RC} = [rc_{i,j}]_{\lvert E\rvert \times \lvert Q\rvert}$

Table 5.1: Summary of Data Fusion Variables for CBMIR

create the final ranking. These variables defined in Chapter 3 are summarized in Table 5.1.

5.1.1 History of Data Fusion

The history of data fusion and Information Retrieval (IR) is long and extensive, incorporating many facets of the retrieval process. Areas of research that fall within combination of evidence for retrieval include the combination of document representations and the combination of queries. For our work we will be concentrating on specifically *late fusion* (Snoek et al., 2006) and what Croft terms "Frameworks for Combining Search System Output" (Croft, 2000). For an examination of other types of data fusion refer to Croft (2000) for an overview.

Early data fusion research began with experimentation into the combination of different retrieval models, document representations and query representations (McGill et al., 1979; Das-Gupta and Katzer, 1983; Saracevic and Kantor, 1988). Research by Belkin et al. (1993, 1995) noted that varying these different factors produced different sets of relevant documents, yet exhibited no major changes in performance metrics. Croft (2000) notes that observations from these early studies suggested that it was beyond the capabilities of a single system to retrieve all the relevant documents for a given query. According to Croft this then resulted in two streams of IR systems being developed, one stream was to create single models which can combine multiple *sources* of evidence such as the INQUERY system based on an inference network (Turtle and Croft, 1991). The alternative stream is the development of systems which effectively combine the outputs of multiple searches from different retrieval models (Fox and Shaw, 1994). Interestingly Croft in his review notes for the task of multimedia retrieval, as different modalities are combined this requires the development of systems which combine the ranking from multiple subsystems (what we would term experts) (Croft, 2000). Two notable exceptions to this however are language modelling approaches for multimedia retrieval (Westerveld, 2004; McDonald, 2005) which implement a generative and discriminative approach respectively. Both of these approaches however still combined visual and text data through the use of weighting schemes, and as such the research presented thus far is equally applicable to these approaches. The Garlic system developed by IBM is an early example of a multimedia information system which combined multiple information systems for retrieval through fuzzy sets (Fagin, 1996).

Broadly speaking we can roughly separate data fusion research into two classes, development of approaches to conduct data fusion and investigations into the data fusion phenomena and why it leads to performance improvement. These two classes are not mutually exclusive and several who developed new approaches often examined why they may have worked, however more research would appear to exist on the development and application of methods for data fusion than why data fusion works. In Chapter 3 we reviewed many of the data fusion operators defined in the literature. This included CombSUM and CombMNZ for score combination (Fox and Shaw, 1994), linear combinations (Bartell et al., 1994; Vogt and Cottrell, 1999), rank aggregation approaches (Aslam and Montague, 2001; Montague and Aslam, 2002) and investigations into score normalization (Montague and Aslam, 2001).

Investigations into the behaviour of data fusion began as referenced earlier with observations about the low overlaps of the documents returned by different ranking models (McGill et al., 1979; Das-Gupta and Katzer, 1983; Saracevic and Kantor, 1988). Belkin et al. (1993) noted that “Different representations of the same query, or of the documents in the database, or different retrieval techniques for the same query, retrieve different sets of documents (both relevant and nonrelevant)”. Lee

(1997) examined this research but contrasted it against the findings of Turtle and Croft (1991) and Saracevic and Kantor (1988), where Turtle *et al.* in experiments combining probabilistic and Boolean retrieval results found that the relevant documents retrieved were shared by both approaches, whilst Saracevic & Kantor found that different query formulations found different documents, but that a document's odds of being judged as relevant increased monotonically as a document appeared in multiple result sets. Lee took these two findings to formulate a new hypothesis for the effectiveness of data fusion:

> "different runs might retrieve similar sets of relevant documents but retrieve different sets of nonrelevant documents" (Lee, 1997).

Testing this hypothesis Lee introduced two evaluation metrics to measure the degree of overlap between relevant documents and nonrelevant documents, termed $R_{overlap}$ and $NR_{overlap}$, which we have defined and used in Chapter 4. Lee (1997) finds that the best result from data fusion was achieved when result sets were combined in which relevant documents had high overlap and low overlap for non-relevant documents. The work of Vogt and Cottrell (1999) confirms Lee's observations by conducting pairwise experiments combining 61 TREC submissions. Vogt and Cottrell term the relevant overlap as the 'Chorus Effect', that multiple retrieval systems return the same relevant documents. Smeaton (1998) conducted pair-wise and triple combinations of retrieval runs from TREC-4 data, and found that from the 9 different classes of retrieval system, only 10 of the 36 pairs examined produced a performance improvement. This unweighted linear combination agreed with earlier findings for text IR of the need to be combining effective retrieval systems to observe a performance increase.

Croft (2000) interprets the findings of the work of Lee and of Vogt and Cottrell as being the result of combination of uncorrelated classifiers. Assuming that the retrieval systems being combined are good, that as the result lists being combined are truncated to 1000 results, and that for a given TREC query there are typically only 100-200 relevant documents, that most good systems will return within the 1000 results the 100-200 relevant documents, but as the 'classifiers' (search systems) are uncorrelated, they will return different sets of nonrelevant documents. Furthermore this emphasizes earlier observations that combinations of independent good search systems, produce gains in performance when fused (Croft, 2000).

The data fusion hypothesis of Lee was critically examined by both McCabe et al. (2001) and Beitzel et al. (2004). Both conducted approaches where various system parameters were held constant whilst varying one aspect, such as the ranking model, stemming, stopping, relevance feedback etc. The work of McCabe *et al.* found that when systemic parameters are held constant, that the combination of vector, probabilistic and Boolean retrieval models did *not* improve performance of retrieval, contrary to previous accepted wisdom. This was further demonstrated by a lack of performance improvement when combining results from TREC-6, 7 and 8 queries which produced high overlaps in both $R_{overlap}$ and $N_{overlap}$, meaning that each of the approaches were returning very similar content. Nevertheless this work found that the overlap coefficients were a good predictor of the potential for performance improvement with data fusion, particularly when systems were combined with weights, such that a poor performing system could be discounted. The combination of a poor system with a good system, using weights where the good system was weighted highly, produced performance increases, lending support to the application of weights for expert combination (McCabe et al., 2001).

Beitzel et al. (2004) like McCabe also conducted experiments where system parameters are held constant to measure the impact of combination of different aspects of retrieval systems. The work of Beitzel *et al.* specifically examined the combination of "highly effective retrieval strategies". Assuming this, Beitzel *et al.* hypothesize that combination of highly effective systems through voting mechanisms like CombMNZ are more likely to harm performance, as the highly effective systems have already been optimized and will rank relevant documents highly, therefore the candidates for promotion up a ranked list are lower ranked common nonrelevant documents as the relevant documents are already highly ranked. They further hypothesize that as constants such as the query and stemming for each retrieval model are held constant, that different models will produce approximately the same set of documents for a query, only the relative ranks of these sets are likely to be different. For highly effective systems Beitzel *et al.* find that the combination of retrieval models (e.g. vector space and probabilistic) hurts performance, rather than helps, whilst the overlap coefficients defined by Lee (1997) provide a poor indicator of potential for improvement through data fusion (Beitzel et al., 2004).

These two results however give credence to the application of weighted data fusion to the task of CBMIR. Given that CBMIR is characterized by the combination of multiple poor retrieval experts (Smeaton et al., 2006), we are unlikely to be combining multiple experts that actually perform consistently well for any set of queries. Furthermore as the work of McCabe shows (and indeed our earlier experiments in this thesis), weighted combination of poor retrieval experts can lead to significant performance improvements.

Therefore the task of data fusion in a CBMIR system is to employ methods which generate an effective weight matrix **RC** such that when applied to the result sets which are then fused, an improvement is made in retrieval performance. The following sections will review methods which can be applied for data fusion and retrieval coefficient estimation.

5.2 Query Independent Weighting

Query Independent Weighting is one of the simplest methods that can be employed when combining retrieval experts. Popular in earlier research in CBMIR (Cooke et al., 2004; Amir et al., 2004; McDonald and Smeaton, 2005; Jeong et al., 1999) because of its simplicity, query independent weighting is an empirical method which requires either a training corpus or domain knowledge of the collection being indexed. Weights are statically assigned for each expert in the system and do not change, regardless of the query being issued. For all queries being processed by the system, this single weight matrix, which weights only experts, is always used.

5.2.1 Expert Weight Matrix

Therefore we define a variant of our weight matrix termed $\mathbf{RC_i}$, which sets only the values of rc_i, meaning that only the experts are weighted. In this variant the weight matrix $\mathbf{RC}$ values of $rc_{i,j}$ vary only when the expert (i, rows) changes, query components (j, columns) are uniformly weighted for each expert, effectively reducing our weighting matrix to a vector. This is illustrated in Equation 5.1.

$$\mathbf{RC_i} = \begin{pmatrix} 0.3 & 0.3 & 0.3 \\ 0.1 & 0.1 & 0.1 \\ 0.7 & 0.7 & 0.7 \\ 0.2 & 0.2 & 0.2 \end{pmatrix} \mapsto \begin{pmatrix} 0.3 \\ 0.1 \\ 0.7 \\ 0.2 \end{pmatrix} \tag{5.1}$$

In the query-independent weighting scheme the creation of the weight matrix $\mathbf{RC_i}$ occurs offline, either as the result of a training phase, or with domain knowledge. Only one instance of $\mathbf{RC_i}$ is created for the system. An illustrative example of this process would be weight optimization for participation in TRECVID. A research group participating in TRECVID 2004, would have the TRECVID 2003 collection, queries and relevance assessments for use as training data. The group would set initial weights for their experts and perform batch retrieval runs on the TRECVID 2003 training data whilst examining the MAP scores being generated. Expert weights would then be modified so as to improve the MAP score on the 2003 collection. Once complete, these weights form the final set of weights to be used for experimentation on the TRECVID 2004 corpus (Cooke et al., 2004). As this optimization process used multiple queries, the expectation is that the weight set generated should not be optimized for any single query, but rather be a set that produced decent performance on a range of queries. As a concrete example, given our example system (Table 3.1), static weights could be set as: text expert (0.5), colour expert (0.3), edge expert (0.15) and texture expert (0.15). In that setup the

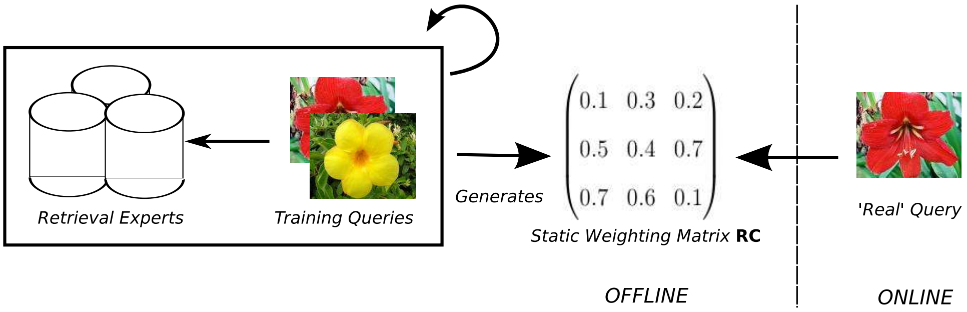

Retrieval Experts
Training Queries
Generates
$\begin{pmatrix} 0.1 & 0.3 & 0.2 \\ 0.5 & 0.4 & 0.7 \\ 0.7 & 0.6 & 0.1 \end{pmatrix}$
Static Weighting Matrix **RC**
'Real' Query
OFFLINE
ONLINE

Figure 5.1: Query Independent Weighting

primary expert would be the text expert, followed by colour, then edge and texture. These weights would remain constant for all queries to the system. Figure 5.1 illustrates this scheme.

The advantages of this approach are that implementation is straightforward, weights are statically assigned, therefore when results from a retrieval expert are returned the weight can be directly applied and the experts combined. This also results in the fastest type of weight assignment that can occur, as the weights are set prior to the system accepting queries. Acceptable performance can be obtained using this method in fairly constrained retrieval scenarios. For instance, if the CBMIR system has indexed a very specific content domain, such as X-Ray images, and a domain expert is available with representative queries, a good static combination of experts can be found. This is an example of a narrow domain, as opposed to the broad domain in which we are operating in, and is an example of where CBMIR has been effective (Smeulders et al., 2000).

There are several disadvantages to the use of query independent weighting. Firstly, the collection being indexed needs to be relatively homogeneous, such that a generalized set of weights can be deployed. Coupled with this, the training queries would need to be representative of what the operational system may have to deal with. However as there are only one set of weights being used, either the range of queries the system will handle is anticipated to be quite narrow and therefore good weights can be set for this 'class' of query, or the system will have to deploy a ‘general’ class of weighting which would hopefully achieve moderate performance on a range of queries. Secondly, issues of overfitting on the training collection are difficult to avoid when using this approach, particularly if the optimization metric being used is MAP. An example of this is found in the TRECVID (Smeaton et al., 2006) benchmarking campaign. Taking the 2004 campaign (Kraaij et al., 2004), we see that three of the top four performing search topics are sports topics (ice hockey, golf and tennis). These queries can be quite different to other queries against a news video collection. Typically they perform well with colour features (e.g. ice hockey will have huge swathes of white in a frame), whilst the associated speech may not be as helpful, so a weighting set would upweight colour and downweight text. Whilst this set of weights will perform well for these sports topics, it may do poorly with the other 21 topics in the collection. A problem arises in that if the optimization done is using MAP, then the MAP value will be dominated by these three high performing queries, and as such, optimizing weights on MAP is in effect optimizing weights on these high scoring topics, resulting in a weight set that is heavily skewed to a subset of potential topics the system may handle. In effect, this subset of sports queries can be viewed as a distinct ‘class’ of query. The next approach we examine implements this approach of defining ‘query classes’ and optimizing on each.

5.3 Query Class Weighting

Query-Class weighting can be seen as an evolution of query independent weighting as it directly addresses many of the failings of query independent weighting and as such has proven popular in the multimedia community. The central concept of this approach is that given a training collection and an appropriate set of training queries, query clusters (i.e. *query-classes*) can be found such that queries within each cluster share some similar ‘properties’ which differentiate them from other queries in the collection (where ‘properties’ may be artefacts such as semantic similarity, performance similarity, distance etc). By partitioning a set of training queries into

Query Independent Weighting Summary	
Weighting Granularity	$\lvert E\rvert$ for all queries (i.e. one instance of $\mathbf{RC_i}$)
Pros	Simple to implement. Once training complete, very fast as no additional query time computation. Appropriate for narrow domains and associated queries.
Cons	Performance poor for more generalized collections and/or queries. Requires adequate training content which captures likely query space. Very prone to overfitting problems, particularly if optimization metric is MAP.

Table 5.2: Query Independent Weighting Summary

discrete *query classes*, it is then possible to optimize for each *query-class* an instance of the weighting matrix **RC**, such that each class should have a different set of weights for combining retrieval experts. When a test/live query is then processed, it is first mapped to a *query-class* so that the relevant matrix **RC** can be applied for the retrieval experts used. As such, this provides a considerable improvement over query independent weighting, notably in the granularity of how experts can be weighted as every *query-class* has an associated weight matrix **RC**. Whilst the query independent approach only has one instance of **RC** to weight queries, in the *query-class* approach there are |*query-classes*| available for weighting. However like the query independent approach, our weighting matrix only weights at the expert level, not at the level of $\langle expert_i, query_j\rangle$, therefore we refer to these matrices again as $\mathbf{RC_i}$. Figure 5.2 provides a high level overview of the *query-class* approach.

5.3.1 Early Approaches

The *query-class* approach was initially developed by Yan et al. (2004) closely followed by Chua et al. (2004) for content-based video retrieval on the TRECVID corpora. Both share a degree of similarity in implementation as both statically define *a priori*

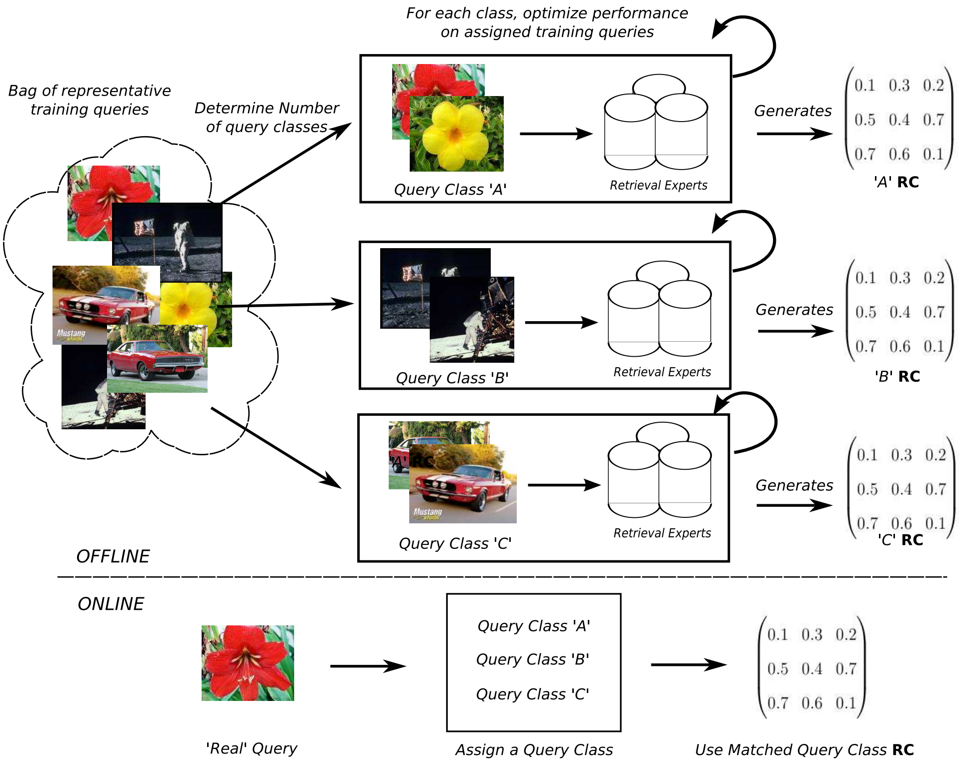


Figure 5.2: Query Class Weighting

the *query-classes* to be used by the system. Yan *et al.* define four *query-classes* based on the intent of the query (providing perhaps a degree of generality to the chosen classes), namely "Named Person" (find shots of Ronald Reagan), "Named Object" (find shots of the White House), "General Object" (find shots of cats) and "Scene" (find shots of beaches) (Yan et al., 2004). Chua *et al.* define six classes, however these are based on perceived information needs of searching within broadcast news video and observations from previous TRECVID queries. Those six classes were "Person" (find shots of Ronald Reagan), "Sports" (find shots of ice hockey), "Finance" (find shots of stock graphs), "Weather" (find shots with rain), "Disaster" (find shots of flooding) and "General" (catch-all for everything else) (Chua et al., 2004). For these approaches to be successful a mechanism is required to assign test queries to the predetermined query classes.

Both approaches analyse the text component of any given query to determine its assignment to a *query-class* via a heuristic framework. Yan et al. (2004) first conduct named entity extraction which assigns queries to either of the ‘Named’ classes, followed by part-of-speech (POS) tagging, noun phrase (NP) and verb phrase (VP) identification with syntactic parsing. They assign queries with one longest matched NP to the “General Object” class, whilst those with more than one NP are assigned to the “Scene” class. Similarly Chua et al. (2004) also perform named entity extraction for query assignment to the “Person” class, however for the remaining classes (except “General”), a set of keywords for each class is defined and queries which contain those keywords are assigned to the matching class. If no keyword matches are found which map a query to a class, then the query is assigned to the “General” class. Once *query-classes* have been populated with training examples, the weighting matrix $\mathbf{RC_i}$ can be set. Chua et al. (2004) employ domain knowledge for weight selection, whilst Yan et al. (2004) treat the task as a maximum likelihood estimation problem and utilize the Expectation Maximization (EM) algorithm (Whitten and Frank, 2005). Yan et al. (2004) also implement a hierarchy when determining the weights for expert combination, firstly combining visual experts into a single result, then combining the aggregated visual expert with the text expert (illustrated in Figure 5.3). This was based on the examination of previous TRECVID evaluations (pre 2004) that indicated that performance was driven by text retrieval on the speech or transcription content of the video (Hauptmann and Christel, 2004), however in more recent TRECVID evaluations this would no longer appear to be the case (Over et al., 2007, 2008). Nevertheless both of these approaches demonstrated improvement over query independent weighting strategies for the corpora on which they were tested.

There are several issues with this initial implementation which posed problems for system builders and users. Firstly, the definition of the *query-classes* is performed manually, potentially with the assistance of a domain expert, typically by manually examining a set of training queries and performing some sort of partitioning of the training space. This partitioning being a manual process implicitly captures any bias that the human performing the partitioning may impart. The result is illustrated

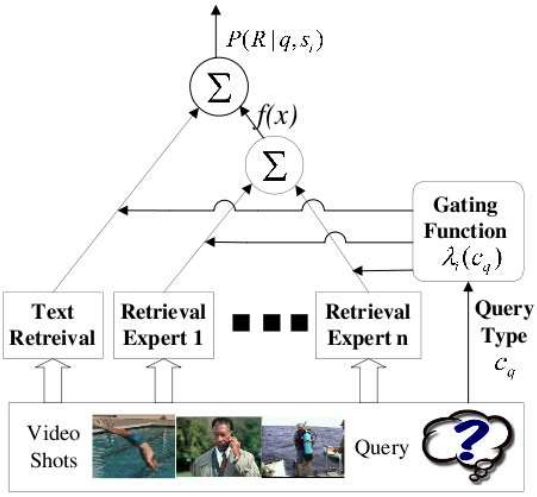


Figure 5.3: CMU Query Class Combination Hierarchy (Yan et al., 2004)

by the two approaches previously discussed, where of the defined classes there is only one in common to both approaches ("Named Person") (Kennedy et al., 2008). Secondly from a user's perspective, the interpretation of the meaning of the label of the *query-class* can be misaligned between a user's interpretation of the label and what the system designer intended the label to represent (Saracevic, 2007; Santini and Dumitrescu, 2008). Finally as this manual grouping is conducted using intuition and perceived topicality, it makes the assumption that queries which belong to the same class will in fact perform to similar levels to other queries in that class and will benefit from the same weighting combination. For instance the 'General' class previously defined would need to employ relatively generic set of expert weights, as it is a catch-all for any query not assigned to a specific class (Kennedy et al., 2005).

5.3.2 Current Approaches

Kennedy et al. (2005) extend the initial approaches by automatically discovering *query classes* in a training set, free of any manual involvement. Their approach is based on the observation that the intent of *query class* weighting is to group queries together such that they share the same set of expert weights to achieve good performance. Therefore Kennedy *et al.* hypothesize that given a set of retrieval experts and training queries, that those queries which share similar performance variations between experts (where performance is measured by an evaluation metric, i.e. average precision) should belong to the same *query class.* For example, if we have available in our CBMIR system a text and colour expert, those queries which produce good performance from the text expert, but poor performance from the colour expert should be clustered together into the same class (Kennedy et al., 2005). Kennedy *et al.* refer to this as clustering in the *Performance Space.*

Performance Space clustering however is dependant upon ground truth data (i.e. relevance judgements) being available for clustering the search queries, and is therefore only applicable to clustering training queries, as for test queries there is no ground truth data available. To address this issue they also define a *Semantic Space* in which to cluster, which similar to the previously discussed methods utilizes various Natural Language Processing (NLP) techniques (named entity, POS tagging etc). The task becomes to align these two spaces such that *query classes* (i.e. query clusters) retain some consistency both with the clustering in the *Performance Space* and with the *Semantic Space.* Kennedy *et al.* define a *Joint Performance/Semantic Space*, matrix D which is a weighted summation of pairwise distance matrices of the *Performance Space* and the *Semantic Space.* With this combined matrix created, clustering occurs within it (either K-means or Hierarchical Clustering (Whitten and Frank, 2005)) to discover the final *query classes*, with the selection of the number of clusters arrived at through empirical testing. Once the classes have been established and populated with training queries, each class can then be optimized to produce the weighting matrix $\mathbf{RC_i}$, in a similar fashion to previous implementations. This approach demonstrated improvements over the aforementioned manual class creation techniques (Kennedy et al., 2005).

Yan and Hauptmann (2006) like Kennedy *et al.*, extend their earlier work to address the issues present with the first implementation of *query-classes*, notably the creation/assignment of *query-classes.* Yan and Hauptmann define a framework termed “Probabilistic Latent Query Analysis (pLQA)” in which *query classes* are latent variables and the process of *query-class* discovery and parameter tuning for a *query-class* are combined into a single phase. There are several distinguishing features between this and the approach of Kennedy *et al.* The main distinction, is that whilst the work of Kennedy *et al.* defines the *Performance Space* which is based on the retrieval evaluation scores of individual experts, Yan and Hauptmann instead utilize the weights learned for expert combination directly for class discovery. In both the manual *query-class* approach and that of Kennedy *et al.*, the definition of the query classes occurs first, followed by expert optimization of the training data assigned to those classes, whereas in this work of Yan and Hauptmann the *query classes* are expressed as latent variables which can be estimated directly from the training data along with the weights for expert combination. A further advantage of this approach is that the mapping of queries to *query-classes* is a probabilistic membership assignment which allows mixtures of classes to occur (Yan and Hauptmann, 2006).

The performance of this approach demonstrates improvement over both query independent weighting and manual creation of *query-classes* (Yan and Hauptmann, 2006). Of interest however is the reported performance of the manual *query class* creation approach, which whilst being outperformed still achieves good performance. Furthermore, when Yan and Hauptmann utilized larger sets of training data (i.e. pooled queries from the TRECVID benchmarks 2004 and 2005, coupled with 40 additional queries defined in house, optimized on the TRECVID 2004 development collection, in total 88 training queries), the performance of the manual *query class* approach demonstrated performance increases, highlighting the need for adequate and representative training data (Yan and Hauptmann, 2006).

Kennedy et al. (2008) notes that this approach by Yan and Hauptmann is likely to achieve greater improvement over Kennedy’s earlier work based upon the reported evaluation metrics on similar collections. However this observation is difficult to substantiate as direct comparison is hampered for several reasons. Firstly the experts utilized are different, and without reporting on individual expert performance it is difficult to gauge comparable performance increases through expert combination. Secondly the collections utilized have little overlap, particularly with the extended training corpus developed by Yan and Hauptmann, which again makes direct comparison difficult. Whilst it would appear that the approach of Yan and Hauptmann offers certain theoretical advantages over Kennedy's work, it may be that some combination of the two approaches produces further improvement (e.g. transform Kennedy's *Performance Space* from optimizing on retrieval evaluation scores, to clustering based on the similarity of weights for class creation, similar in spirit to that of Yan and Hauptmann).

One of the latest evolutions in the *query class* approach is provided by Xie et al. (2007), whose work concentrates on creating dynamic *query-class* weighting during system operation, rather than the previously described approaches which learned classes from training data. The approach by Xie *et al.* like the majority of the previous approaches, processes the text component of a query to arrive at the *query-classes.* Xie *et al.* first use the PIQUANT engine to tag the query text, using semantic tags from a broad ontology of over 100 concepts designed for intelligence and news domains with question answering applications (Xie et al., 2007). As the tags relate to semantic concepts, they need to be mapped to a visual domain for video retrieval, as such Xie *et al.* define seven 'features' in a 'semantic query feature space' which are: "Sports", "Named-Person", "Unnamed-Person", "Vehicle", "Event", "Scene" and "Others".

From these seven 'query features' they construct three approaches, two of which emulate the previously discussed approaches plus their own approach. *Qclass* is the standard *query-class* approach, where each of the seven 'query features' becomes a *query-class*, training queries are mapped to one of the seven and then weights are learned from these to provide the weights for the class (i.e. generation of matrix $\mathbf{RC_i}$ for each class). To process an unseen query, the query is mapped to a class, and the previously optimized weights for that class are used.

The second approach *Qcomp* is similar in intent to the work of Yan and Hauptmann (2006), as each 'query feature' becomes a 'query component', where a query may have membership with multiple 'query components'. Once the training queries have been mapped to 'query components', the weights are learned through the usual optimization process. Processing an unseen query involves determining the 'query components', then taking the average of the weights for each of the components matched for the unseen query. That is, each 'query component' will have generated from its mapped training queries, a weighting matrix $\mathbf{RC_i}$. When an unseen query is processed, the final weighting matrix to be used will be the average matrix of each of the matched 'query components'.

Finally, Xie *et al.* define their own approach which introduces a variation to the *Qcomp* approach, which they define *Qdyn.* Like *Qcomp*, queries are parsed and their 'query components' identified, however once these are found a nearest neighbour matching is performed to identify similar training queries. These training queries are then dynamically optimized to produce the weighting matrix $\mathbf{RC_i}$ to be used for the unseen query, producing the 'dynamic query class' (Xie et al., 2007).

In their evaluation, they found that both *Qcomp* and *Qdyn generally* outperformed query-independent weighting and the standard *Qclass* approach, with minimal variation between the two. What is of interest in their experimentation is the high performance achieved by the query independent weighting approaches (Xie et al., 2007). Xie *et al.* account for this as an artefact of sports queries in the TRECVID search corpus (previously discussed in Section 5.2). The approach defined by Xie *et al.* performs more robust semantic processing of the query than in previous approaches, as an ontology of over 100 semantic concepts is utilized, it is in effect only clustering queries on the query text, as opposed to the approaches of Kennedy and Yan which take into account how each the queries cluster in an empirical space.

5.3.3 Observations

In this section we have presented techniques for *query-class* weighting, each of which produces for every class, a weighting matrix $\mathbf{RC_i}$. All presented techniques in this section demonstrated performance improvements when compared against query independent weighting. This is not surprising, given that the granularity of weighting possible with the *query-class* approach is greater than that of the query independent approach, with a separate weighting matrix for each class defined. This approach has also proven popular in the research community, with several groups in TRECVID making regular use of this approach for expert combination.

Nevertheless there are several issues with this approach. Firstly, it is interesting to note that across all the different approaches examined, the reported number of *query-classes* created did not vary greatly, with the range being between 4 - 7 classes. Even in the case of the dynamic approaches such as Yan and Hauptmann (2006) using an augmented training collection which contained 88 search topics, only 6 *query-classes* were used. This is likely an artefact of the utilization of single TRECVID corpora for training in many of the approaches, which typically contain only 24 topics, however general questions remain about the capacity for correctly weighting unseen queries which have no representative in the training corpora. This problem's become particularly apparent in activities such as the TRECVID benchmarking activity when entirely new corpora are introduced for which no training queries with relevance judgements exist.

Secondly, the approaches as described here contained some level of expert aggregation, such that the generation of any weighting matrices were restricted to weighting at only the expert level ($\mathbf{RC_i}$). As we have demonstrated in Chapter 3, by not considering the full set of result sets to be combined, an artificial handicap is placed upon performance.

Third, the majority of the approaches conducted classification on only the query text associated with a query, the rationale for this is given as "query text accurately describes the information need in a multimedia query, it also serves as the sufficient

Query Class Weighting Summary	
Weighting Granularity	$\lvert E\rvert$ (i.e. $\mathbf{RC_i}$) for every query class
Pros	Performance improvement over query independent. Number of classes variable. Classes can be dynamically created after initialization with training data. Dependant on approach, can have fast runtime execution (pre learned classes).
Cons	Weighting restricted to $\mathbf{RC_i}$ for each class. Requires adequate training content which captures likely query classes. Approaches dependant on accurate text processing of the query text. Requires that the query contain text.

Table 5.3: Query Class Weighting Summary

criteria for the human judging on whether or not a query is relevant" (Xie et al., 2007). Query disambiguation is a major research problem in itself, whilst query text provides to the system a more easily translated statement of user intent, caution should be applied in relying solely on it when the user provides other forms of information as part of a query. An example of this is illustrated in the TRECVID 2007 benchmark with the ambiguous query "(0199) Find shots of a person walking or riding a bicycle", which by query text alone is relatively ambiguous (e.g. shots of people walking OR riding, or shots of people pushing or riding a bicycle), and lead to confusion among participants, however the visual query components helped to provide clarity (Over et al., 2007). Furthermore, given corpora such as TRECVID which incorporates multilingual sources of information, the analysis of the query text may become far noisier if it needs to undergo machine translation.

5.4 Machine Learning Methods

Machine learning applications can be considered as techniques which discover structural patterns in data that allow us to explain and make predictions from that data (Whitten and Frank, 2005). They provide us with tools to help make sense of and to organize data. The field of machine learning is incredibly broad and a general review is outside the scope of this work. In this section, we will restrict ourselves to a brief review of approaches which can be characterized as *supervised* and *discriminative*, that is approaches which require training data which has been manually labelled, and models which learn directly on the data provided to them.

The power of machine learning approaches stems from being able to learn patterns within very large volumes of data. However because of this, careful attention needs to be provided over what data is given to a machine learning algorithm as any method is only as effective as the data it leverages. This presents several significant challenges in the application of machine learning approaches to search and data fusion. As a toy example, consider a general image collection of photographs and we wish to find all images of cars. If the collection size is 100 images and the collection only contains two instances of cars, then a classifier could easily be produced which would annotate the entire collection as containing no cars and have achieved an accuracy of 98%, yet still be completely useless to us for the purposes of search, as it provides no information to the user as to which images are relevant for their information need. From the perspective of the learning algorithm it has learnt an approach which is highly successful, yet illustrates the need for careful data preparation, algorithmic selection and optimization/evaluation criteria.

This example highlights one of the major hurdles for the application of supervised machine learning methods to tasks like search and data fusion. That is, the classifier was provided with positive examples, such as an image of a car, of what we are trying to find. However discriminative approaches typically also require adequate negative examples, that is things that we do not want as part of our information need. Section

5.4.1 provides approaches for multimedia retrieval that generate pseudo-negative examples to assist in this problem. Section 5.4.2 proposes a different approach, typically applied in text retrieval, which leverages very large amounts of annotated training data (queries and relevance judgements) to learn what makes documents relevant to a query and as such produce a generalized ranking model.

Other major problems which apply to the use of machine learning approaches are the *class imbalance* problem and the *curse of dimensionality* (Akbani et al., 2004). These two problems are interrelated for the application of search. Put simply, the class imbalance problem for search is that often there are far more irrelevant than relevant items in a corpus, hampering classification, as the classifier can be biased towards the larger of the imbalanced classes. Similarly, when learning from data, such as using visual feature vectors as described in chapter 2, that as the size of the feature vector grows, an exponential increase occurs in the size of the feature space, correspondingly requiring an increase in the number of examples we require in order to achieve a stable model (Bellman, 1961; Beyer et al., 1999; Tešić et al., 2007a).

5.4.1 Generation of Pseudo-Negative Examples for Search

Natsev et al. (2005) explicitly addresses the issue of multi-example, multi-feature search, such as the scenario defined by our example CBMIR system and query (Section 3.1.6), making use of machine learning methods, specifically the use of Support Vector Machines (SVM) (Vapnik, 1995). Natsev *et al.* conduct retrieval experiments using the TRECVID 2003 corpus and topics, where each topic contains not only a statement of the information need but also relevant visual examples (see Chapter 2 for further detail). However, as traditional SVMs conduct discriminative classification, they require not only positive training samples (e.g. the visual example queries) but also negative training samples, which are rarely provided during initial query formulation (the exception is iterative feedback from a user, see Section 5.6.1 regarding Relevance Feedback). Natsev *et al.* propose several mechanisms for generating 'pseudo-negative' examples, however the best approach was a random sampling of the collection. As TRECVID search topics are typically relatively complex information needs (e.g. find shots of George Bush walking versus a classification task of find outdoor shots), the actual number of relevant shots in a collection is relatively small.

Empirically Natsev *et al.* select 50 as the number of pseudo-negative examples to sample per topic, whilst the number of positive examples is typically 5-6. To improve retrieval performance, they repeat the selection of the pseudo-negative examples whilst keeping the positive examples 10 times per query, each iteration producing a model, all of which they statistically average and then merge into a single model. Natsev *et al.* refer to this as *bagging*, however it distinguishes itself from traditional machine learning *bagging* approaches as only the negative examples are randomly sampled, whereas in traditional *bagging* approaches, both classes are randomly sampled and multiple classifiers are built and averaged from these random samples. Therefore for each query, a modified bagging approach is conducted, where 10 'bags' are defined, each of which shares the same positive examples, but a random selection of negative examples.

This learning process is conducted *per expert*. In their work they had 4 visual experts available and through prior empirical testing they select the top 3 performing experts to use for any particular query. The fusion of retrieval experts is through statistical normalization (Z-score normalization), followed by unweighted linear combination. Each of the experts when trained implicitly combines the multiple examples into the learned model, and as such they can be seen to be individually weighted, however the final combination of the retrieval experts uses uniform weighting. Natsev et al. (2005) highlight that performance gains could be made through the adoption of techniques such as query-class weighting (Section 5.3) to appropriately combine expert outputs. Furthermore, as an SVM is utilized in this process, a training phase with appropriate data was required for SVM parameter tuning. As there was no appropriate training data for the search topics, this parameter tuning was query-independent. Nevertheless this approach has proven popular in the

TRECVID community, recently used by both MediaMill (Snoek et al., 2008) and Microsoft Research Asia (Mei et al., 2008) in TRECVID 2008. This approach was also adopted by Tešić et al. (2007b) for semantic video search.

Tešić et al. (2007a) also address the issue of imbalanced and sparse learning instances for visual retrieval in multimedia search, by extending the work of Natsev et al. (2005). Tešić *et al.* examine the modified bagging approach of Natsev et al. (2005) offering improvements to both the bagging approach itself and to the selection of the pseudo-negative examples. The number of pseudo-negative samples is dictated by the bagging parameters $K \times N$, where K is the number of bags to use, and N is the depth of each bag. Natsev et al. (2005) kept the number of bags (K) fixed to 10, whilst the depth of each bag (N) was 50 random pseudo-negative examples plus the positive query examples (P). Empirically Tešić *et al.* again selected $K = 10$, however the value of N became a function of the number of positive examples P. Implementing the work of Akbani et al. (2004) they define the sample of negative examples to be $N = 10 \times P$, and in experimentation found that this ratio offered improvement for more difficult topics, whilst not over-sampling the negative space.

Tešić *et al.* improve upon random sampling for pseudo-negative training examples. Using k-means clustering (Whitten and Frank, 2005) where the number of clusters to be found is defined as $k = 2 \times N \times K$, they randomly select the centroids of N of the clusters. This approach provides a more representative sampling of the negative example space, as each of the selected pseudo-negative examples belongs to a different cluster. Further experiments found a similar approach could be applied to the positive examples by clustering around the positive example space, thus generating additional pseudo-positive examples. Tešić et al. (2007a) found this approach offered improvement over the pure random selection approach proposed by Natsev et al. (2005).

5.4.2 Learning To Rank

Learning to rank is a rapidly expanding area of research which is seeing an influx of machine learning researchers and machine learning methods tackling the ranking problem, most frequently within the domain of text information retrieval. Such is its rapid growth that the last three SIGIR conferences have featured learning to rank workshops (2007-2009). Learning to rank is a catch-all term that encompasses machine learning approaches, typically discriminative, which seek to learn a ranking model given example queries, documents and relevance assessments or implicit user judgements (such as click logs). Their discriminative power comes directly from having massive amounts of data (certainly in comparison to CBMIR) in which to sample and learn (Geng et al., 2007; Xia et al., 2008).

There are fundamental differences between classification tasks and ranking tasks, and many machine learning methods cannot be applied directly to the ranking problem without modification. The major differences between ranking and classification are that in ranking, the order of instances is important, whilst in classification they are not. Secondly, the evaluation metrics differ as ranking places a higher emphasis on ranked precision rather than recall, whilst in classification both recall and precision are equally important and as the ordering of instances is not important all classification errors are equally important (Geng et al., 2007).

Broadly speaking, learning to rank approaches can be classified into three subtypes; *Pointwise*, *Pairwise* and *Listwise* (Liu, 2008; Xia et al., 2008). The approaches primarily differ on what is used for training data and relevance assessments. The pointwise approach treats the problem as similar to regression or classification problems where each sample (single document), is treated as independent and relevance assessments have binary values and are considered absolute. The pairwise approach as the name suggests, randomly samples pairs of documents from training ranked lists, and learns the relative difference in relevance between the two documents. This approach can utilize non-binary relevance judgements, using metrics such as NDCG. These previous approaches learn at the document, or document pair level. The most recent iteration developed is the listwise approach which rather than examining single documents or document pairs, takes as the learning instance a ranked list of documents associated with a query and the respective relevance information, and as such is better able to capture the properties of the query which generated the ranked list (Cao et al., 2007; Geng et al., 2007; Liu, 2008; Xia et al., 2008; Geng et al., 2008; Liu et al., 2009).

The attraction of learning to rank approaches is their ability to learn ranking models which incorporate large numbers of document features. For example, the LEarning TO Rank (LETOR) dataset, provided by Microsoft Research Asia (Liu et al., 2007a), provides 44 extracted features per document for the .gov document collection from which models can be learned, including both 'low-level' features such as term frequency through to 'high-level' features such as BM25 scores. Commercial search engines in implementing learning to rank approaches take this number far higher, providing at least 600 features for learning a ranking model (Cao et al., 2007). Typically the product of the learning is a single, very well generalized, ranking model which successfully incorporates all the feature data, therefore producing an effective query-independent model (Geng et al., 2008). In order however to leverage these massive amounts of features and build an accurate ranking model, equally large numbers of training queries and relevance assessments are required. For the .gov LETOR collection, 450 queries and relevance assessments are available, whilst the work of Cao et al. (2007) using commercial search data had 25,000 queries leveraged.

Whilst no doubt effective, these approaches face significant hurdles in their translation to CBMIR applications and data fusion. Firstly, in the text benchmark datasets it is often only the queries which are partitioned into training and test sets, whilst the document data itself remains unchanged. This contrasts to CBMIR benchmarking datasets such as TRECVID, where not only are the search queries partitioned into training and test sets, but the data itself is also partitioned between training and test, illustrated in Figure 5.4. This significantly complicates learning approaches, as when the document data is static, models are able to learn *for that* *collection* what features impact upon relevance. When the documents themselves also change, as well as the queries, the generated models at best *estimate* what features impact upon relevance, assuming that the training documents are an accurate sample of the eventual test documents. Whilst this later problem is common in classification tasks, the difference is that for classification tasks, the 'query' does not change between the training and test collections, whereas for search tasks on TRECVID data both the queries *and* the data changes between training and test activities. In addition, to learn a generalized model which is not overfitted to any particular type of query, a very large amount of training data and associated relevance assessments are required. Typically this is not available for CBMIR benchmarking datasets. Nevertheless learning to rank techniques have been applied to tasks of multimedia and query-dependant search, which we briefly review.

5.4.2.1 Multimedia, Expert Combination and Query-Dependent Applications of Learning To Rank

Liu et al. (2008b) propose a pointwise approach for 'query-independent' learning for video retrieval, contrasting it to the methods developed by Natsev et al. (2005). Liu *et al.* utilize a pointwise approach for TRECVID data where mining either relevance judgements or search logs, they associate queries with individual video shots along with a relevance judgement, from which they mine various textual and visual features. Their experiments demonstrate a performance gain over the query-dependent approach of Natsev et al. (2005), however their documentation of their implementation of that approach is sparse, and performance gains could be due to the use of features which are biased towards their model. The motivation for the approach is that a generalized ranking model can be learnt which the avoids overfitting problems of query specific learning approaches. This motivation is contrary to our findings in Chapter 4.

Liu et al. (2007b) tackle the issue of metasearch and rank aggregation for expert combination by proposing a 'Supervised Rank Aggregation' approach. Utilizing the

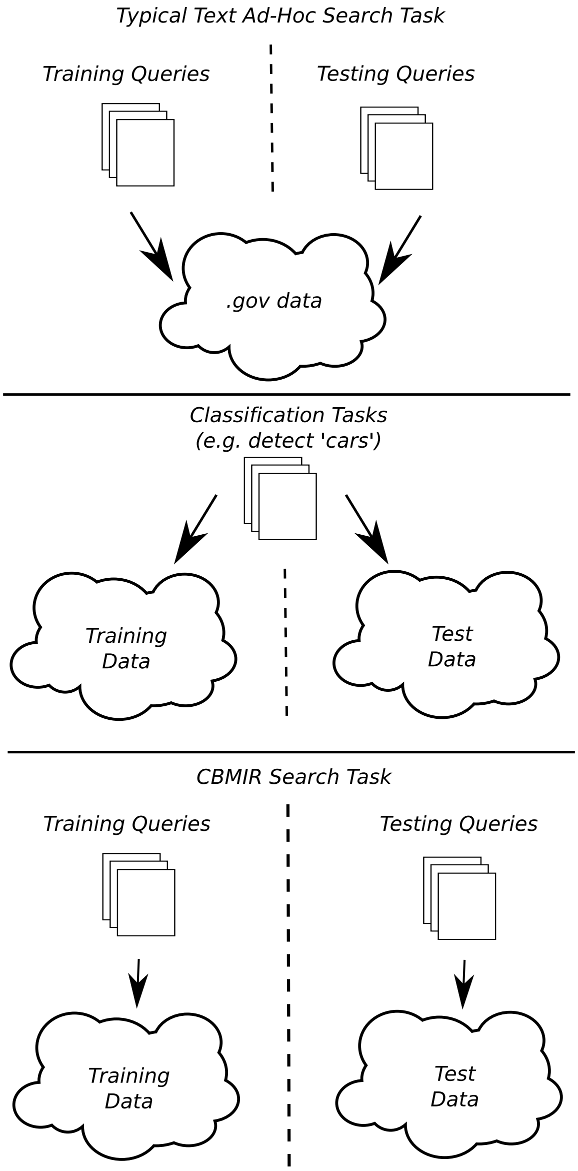


Figure 5.4: Differences between tasks and training/test partitioning; text ad-hoc search, traditional classification, CBMIR search

relative ranks of training result sets and associated relevance judgements, they learn a weighted Markov chain to combine the top-100 results of six commercial search engines. Their experiment utilized 500 queries mined from a commercial query-log. Interestingly they found that the weight assigned to a search engine through this process did not correlate between search engines which returned similar documents. That is if search engine a and search engine b returned similar documents, a may receive a high weight whilst b does not. Conversely if a search engine c returns dissimilar documents to a and b it was given a high weight. The result appears counter intuitive to the data fusion hypothesis (Lee, 1997), and is perhaps a result of a generalized model being learned over incomplete document collections with a large range of queries.

Geng et al. (2008) highlight the problem of previous approaches to learning to rank, which build a generalized model from a very large amount of training data, that specific types of queries will perform better with a more specific rather than generalized ranking model (thus in direct opposition to the work of Liu et al. (2008b)). Geng *et al.* propose a K-Nearest Neighbour (KNN) approach which performs 'soft' classification of a test query into the query space, which is populated with an expansive set of training queries. The distinction between 'soft' and 'hard' is that 'hard' classification would classify a query into query clusters defined *a priori* whereas 'soft' classification projects into the query feature space and dynamically defines a nearest neighbour query cluster, very similar to the approach of Xie et al. (2007) (despite the assertion by Geng *et al.* of no prior relevant work). Geng *et al.* then utilize a pairwise learning to rank approach, specifically Ranking SVM (Joachims, 2002), to learn the ranking model for the dynamically created query cluster. Like all the previous learning to rank approaches, this work leverages very large amount of training data with relevance judgements. Specifically when defining the dynamic query cluster using KNN, the value of k queries to assign to the cluster from which to learn the ranking model, was varied between 100 and 1500 training queries with optimal performance achieved at $k = 300 - 800$.

5.4.3 Observations

Unquestionably the influence of machine learning methods in Information Retrieval is rapidly expanding, the performance of these methods offering significant improvement over unsupervised methods (Liu, 2008). The ability to learn from data which incorporates vast numbers of features, whilst utilizing massive amounts of training data, produces discriminative models that offer 'unreasonable effectiveness' (Halevy et al., 2009). Often the proof of effectiveness is in an approaches uptake and popularity, anecdotally it would seem that the major commercial search engines employ these sorts of machine learning methods for ranking web documents (Liu et al., 2007b; Halevy et al., 2009). Whilst these approaches seem likely to be the future of search, their current application to Content-Based Multimedia search and data fusion offers significant challenges.

The main challenge is a lack of large amounts of training data, specifically training queries and relevance judgements. The aforementioned approaches success is largely due to the exploitation of massive quantities of explicit and implicit training data, such as relevance assessments or click-logs, which on a large scale reveal relationships between relevance and the features that make something relevant. Coupled with this is the dimensionality curse. Whilst text based approaches used over 600 features per document on which to train, many of these features could be regarded as 'higher-level' such as BM25 scores, whereas content-based features are often values such as histogram bins and operate on a much lower level of abstraction. Not reviewed here, but some multimedia practioners avoid some of these problems by incorporating semantic concept detection (see Chapter 2) as features for training, such as Tešić et al. (2007b). This offers some way forward but shifts retrieval problems into issues of query-independent concept detection and the adequacy of concept ontologies coverage over the potential search space.

As content-based multimedia retrieval continues to become more mainstream, approaches such as learning to rank can be re-evaluated for their applicability as training data becomes more common place. This does not mean that multimedia

	Machine Learning Summary
Weighting Granularity	Dependant on approach, $\mathbf{RC_i}$ through to $\mathbf{RC_{i,j}}$
Pros	Potential weighting of pairs $\langle expert_i, query_j \rangle$. Learns from data. Can produce very effective generalized models. Pseudo sampling approaches allow for runtime generation of discriminative models of an information need.
Cons	Requires *significant* levels of training data. Must handle class imbalance problem and the curse of dimensionality. Learning to rank approaches typically work on the same corpus of documents for training and test (Figure 5.4).

Table 5.4: Machine Learning Summary

search will then become a solved problem. These approaches only provide a framework in which generalized models can be learned for retrieval, they are only as good as the data on which they are learned. Considerable effort can be spent in the development of attributes which can be found to populate feature vectors for training (e.g. a CBMIR equivalent score for BM25) from which a machine can learn. Finally, many of these approaches produced very good *generalized* retrieval models, yet we have seen from Chapters 3 and 4 the performance of correctly weighted individual pairs $\langle expert_i, query_j \rangle$ indicates the need for query-dependent weighting.

5.5 Score Distribution Methods

In our definition of the terms used in this thesis, we stated that a $document_m \mapsto (name, rank, score)$, that is for every retrieval expert used we assume it is under our control and that we have full access to not only the ranked list of documents for a query, but also the scores that the retrieval expert assigned to those documents in order to produce that ranking. One class of method which has been proposed for combining the outputs of retrieval experts, is to model the score distributions of ranked lists of retrieval experts. The central idea is that a score distribution for relevant documents and a separate distribution for non-relevant documents can be identified. Once these are known they can be compared against the ranked scores of a retrieval expert, such that inferences can be made as to which expert's ranked list is more likely to contain relevant documents, and therefore combine these lists accordingly.

The idea that two distributions exist which can model relevant and non-relevant document scores has existed for some time as explained by Robertson (2007), beginning with Swets (1963) who proposed two Gaussian distributions of equal variance, and later two exponentials (Swets, 1969). Tangentially, our own proposed approaches for weighted data fusion involve an examination of retrieval expert score distributions, however we do not explicitly model any particular type of distribution (See Chapter 6 for details).

Utilizing score distributions of relevant and non-relevant documents enables system creators to explore multiple applications. Proposed areas of application for this research include expert combination, multi-lingual retrieval, filtering applications, distributed retrieval and topic detection (Manmatha et al., 2001; Arampatzis and van Hameran, 2001).

5.5.1 Current Approaches

Currently, the most popular form of modelling score distributions is the use of an exponential distribution to model non-relevant documents and a Gaussian distribution to model relevant documents. These observations were developed independently by Manmatha et al. (2001) and Arampatzis and van Hameran (2001). Figure 5.5 from the work of Arampatzis and van Hameran (2001) visualizes these distributions for topic 'FT-352' of the TREC-9 filtering task (Ault and Yang, 2002).

The motivation of the work of Arampatzis and van Hameran (2001) is a document filtering task, i.e. to monitor a temporal stream of documents (e.g. news reports) with a persistent information need, and to return to the user relevant documents from that stream as quickly as possible (Ault and Yang, 2002). As such the system

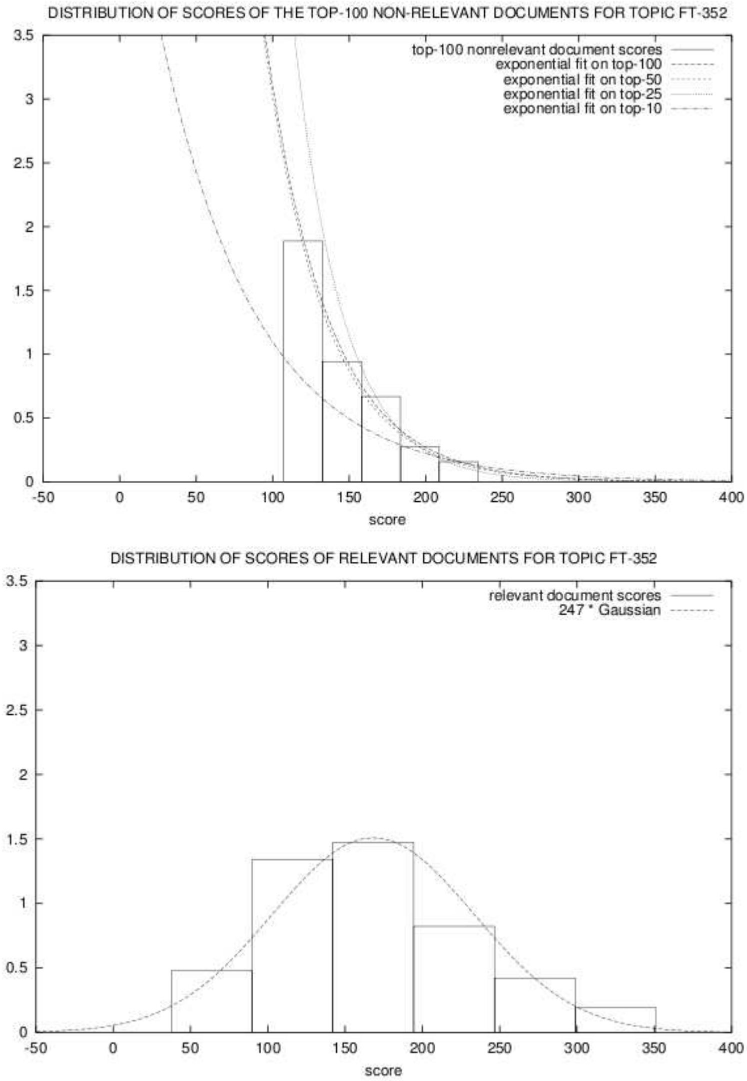


Figure 5.5: Exponential and Gaussian distributions for non-relevant and relevant documents (Arampatzis and van Hameran, 2001)

built is an adaptive binary classifier, where the task is to threshold the scores of retrieved documents, such that scores above the threshold are returned, whilst those below are assumed to be non-relevant and discarded. Using training data with known relevance judgements, Arampatzis *et al.* derived the distributions of relevant and non-relevant document scores and utilized this to maximize a utility function which determined the optimal threshold for returning documents to a user. A Rocchio inspired approach, this work required query expansion of approximately 250 words in length. This approach was later extended to incorporate Maximum Likelihood Estimation for the selection of the threshold parameters (Zhang and Callan, 2001).

Manmatha et al. (2001) also utilize the analysis of score distributions (again with a Gaussian for relevant and exponential for non-relevant), however the application which they tackle directly is the combination of retrieval experts. Manmatha *et al.* perform experiments combining the results of a probabilistic search engine and a vector space search engine. Their work differs from that previously described, as no explicit use of relevance assessments is made in their combinatorial model. Having previously observed the form of the score distributions, Manmatha *et al.* develop a mixture model which consists of both the Gaussian and exponential distributions, and fit this to the score outputs of a retrieval expert using the EM algorithm (Whitten and Frank, 2005), where the task is to identify the mixing parameters and component densities. Once the model has been fitted the scores of a retrieval expert can then be converted into probabilities of the score being relevant.

The results reported by Manmatha et al. (2001) are interesting, showing that experts combined using the mixture model fitted by EM performed on a par with combining experts using CombMNZ (see Chapter 3). When relevance data was incorporated so that the distributions could be fitted directly to the experts, improvement was found, indicating that the fitting of the mixture model without relevance data could be improved. Manmatha *et al.* however list some caveats with their approach. First is the application of EM for fitting the mixture model and the cold-start problem, i.e. the initial set of parameters selected for running EM need to be carefully chosen or else the algorithm may settle on parameters which form a local rather than global maxima. Second, the fitting of the distributions was contingent on the retrieval experts being 'good', they found the distributions a poor fit for retrieval experts which performed below average. Finally the retrieval experts being combined need to perform to a similar level, or else performance is degraded. Implicit in this is that no actual weighting of retrieval experts is performed, the transformation of the scores into probabilities is in itself allowing the experts to be directly combined, if the transformation is accurate enough weighting would not be required.

5.5.2 Theoretical Examination

Robertson (2007) reviews the work conducted in the area of score distribution analysis and conducts a theoretical review of the various approaches, examining each of the proposed distributions of relevant and non-relevant documents to determine their theoretical validity. We review this theoretical examination as the approach of utilising score distributions is one we will follow in the development of our own weight generation algorithms detailed in Chapter 6. Robertson begins with the probability ranking principle, which states that a search system should rank documents in order of their probability of evidence (Robertson, 1977), to formulate the "Convexity hypothesis" which states that:

> "For all good systems, the recall-fallout curve (seen from the ideal point of recall=1, fallout=0) is convex" (Robertson, 2007).

where the 'recall-fallout curve' is a Receiver Operating Characteristic (ROC) curve (Whitten and Frank, 2005) which plots recall against fallout (proportion of relevant documents retrieved against non-relevant documents retrieved, see Chapter 2), shown in Figure 5.6. A straight line on this graph (running from (0,0) to (1,1)) represents a random ordering of documents in response to a query, and therefore identical distributions of scores for relevant and non-relevant documents. If the line

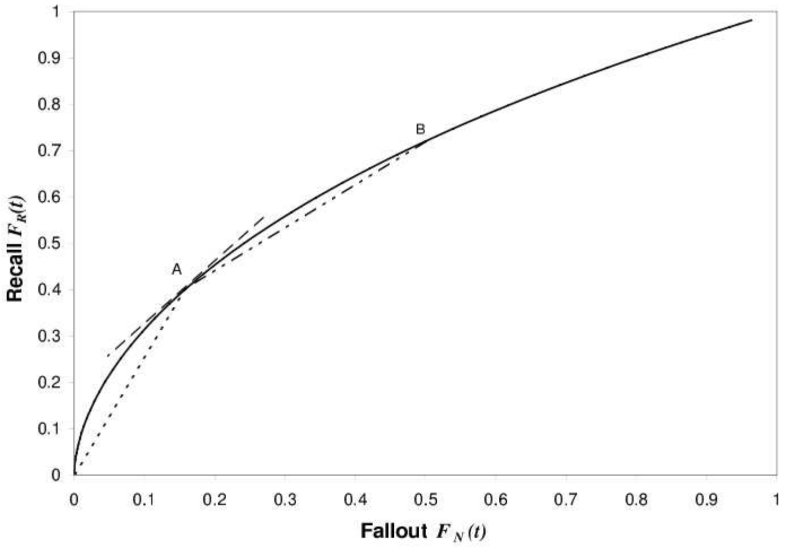


Figure 5.6: Recall-Fallout ROC curve (Robertson, 2007), convex in shape. Straight lines indicate random orderings, deviations from that highlighting biased ordering.

plotted is convex in shape (i.e. biased towards recall), the ranking is good as it is able to distinguish relevant from non-relevant documents. Conversely if the curve, or parts of the curve, are concave, the ranking is poor, and can be trivially improved by randomly reordering the documents such that the curve moves to a straight line (Robertson points out that better than this the ordering could simply be reversed to transition the curve from concave to convex).

Taking the 'convexity hypothesis' and the probability ranking principle, Robertson examines the proposed score distribution pairs of relevance and non-relevance to determine if they violate these hypotheses. Of the earlier proposed distributions, the approach of modelling both relevant and non-relevant as exponentials, and modelling relevant and non-relevant as Gaussian with equal variance does not violate these conditions. However for the case of an exponential for non-relevant and Gaussian for relevant distributions, Robertson found that this combination violated the convexity hypothesis, producing a concavity in the initial rankings, as well as potentially at the end of the ranking. This finding is acknowledged by Manmatha et al. (2001) who also observed this phenomena, which produced non-monotonic probabilities in some of their topics. Their solution to this was to redefine the probability of relevance when this occurred, assigning the top ranked document a probability of 1, then having a linear decrease in probability with the scores until the top of the curve of estimated probabilities is reached, then resuming from that point downwards (Robertson, 2007; Manmatha et al., 2001). The implied justification for this procedure is that the probabilities should be a monotonic function of the score. Robertson‘s alternative to this approach is to randomly re-order those top ranked documents, such that when plotted on the recall-fallout graph a straight line is produced, thus providing a well founded form of extrapolation (Robertson, 2007).

Having established these theoretical inconsistencies, Robertson (2007) further highlights impacting factors, including distributional changes through result set truncation and the issue of score normalization and non-linear transformations (e.g. converting to log-odds probabilities, then re-normalizing to the range [0,1] using a linear transformation), which in themselves may also be producing the phenomena which is being observed. As such, Robertson concludes that whilst empirically the normal/exponential pair might approximate the distributions of relevant and non-relevant documents, the approach violates the convexity hypothesis and calls into question its general validity (Robertson, 2007).

5.5.3 Observations

The nature of score distribution research is appealing, as it provides a mechanism in which to potentially generate weights for every pair $\langle expert_i, query_j \rangle$, such that we can populate the full weighting matrix $\mathbf{RC_{i,j}}$ rather than just populating expert level weights (such as discussed in 5.2 and 5.3). However, the application of this area to the task of CBMIR appears problematic. Leaving aside the theoretical issues raised by Robertson (2007), there are several pre-requisites that CBMIR would appear not to meet. Firstly, as defined by Manmatha et al. (2001), this approach only works well when ‘good’ retrieval experts are used, which as demonstrated in earlier Chapters

Score Distribution Fitting Summary	
Weighting Granularity	$\mathbf{RC_{i,j}}$
Pros	Potential for very fine grained 'weighting'. Demonstrated performance improvement over using individual experts.
Cons	Requires either retrieval experts to be of similar standard, or for verbose query expansion. Dependant on experts processing the same query. Based on empirical observation but theoretically invalid.

Table 5.5: Score Distribution Fitting Summary

for various experts is far from the case. Similarity in the case of Arampatzis and van Hameran (2001), verbose query-expansion was required to illicit the required scores from experts in order to observe the two distributions.

Secondly in the described approaches the experts being combined, whilst having different retrieval algorithms, had the same document representations and as such each expert could be given the exact same query. However when the queries issued to retrieval experts are completely different queries, as in the case of a CBMIR multi-example query with a text and visual component, combining expert results using an absolute measure (such as provided by Manmatha et al. (2001)) presents problems, as the variance and diversity of an expert's results makes absolute comparison problematic. To alleviate this, weighting could be employed to make the results more cross-comparable, but in so doing we are back to our initial problem of having to estimate correct weights for each set of results.

5.6 Other Methods

In this section we will briefly review other methods which have application for data fusion. We list these approaches under 'other methods' as they either occur after the initial retrieval process, or leverage aspects of the underlying data which we cannot for CBMIR.

5.6.1 Relevance Feedback

Relevance Feedback has long been an area of active research within the information science community (Ruthven and Lalmas, 2003)(Zhou and Huang, 2003), dating back nearly 40 years (Rocchio, 1971). The idea behind relevance feedback is that it can be difficult for a user to formulate a query to an Information System which will effectively capture what is being sought, but that a user presented with relevant information will recognize it. A system can utilize these user judgements to better refine the ranking presented to a user, typically in an iterative process (Ruthven and Lalmas, 2003). As such it is a valuable technique for bridging the information need of a user to a form that the system can better exploit.

Typically however, relevance feedback is an interactive process, or at the very least occurs after an initial ranking by an information system. Therefore for the purposes of this thesis, we do not consider relevance feedback as one of the approaches which we will examine in depth, as we are concerned with data fusion with regard to the generation of the best initial ranking. Whilst the study of relevance feedback does have some applicability to data fusion, as often multiple forms of evidence need to be combined, we state that the insights we obtain in this thesis can be applied to relevance feedback as a separate activity in order to improve it, and indeed relevance feedback can be applied to any of our techniques explored here in order to improve retrieval performance.

Zhou and Huang (2003) conduct a thorough review of relevance feedback approaches in image retrieval, whilst Datta et al. (2008) in their general review of image retrieval provides some additional updates to this work.

MARS, an early content-based multimedia retrieval system by Rui et al. (1997), implemented relevance feedback similar to what is found in many text retrieval systems. Utilizing textures as the content-based feature, it uses a vector space representation and investigates both *tf.idf* and Gaussian normalization for determining the weight vector.

Machine Learning approaches feature prominently in image relevance feedback literature, their popularity due in part in being able to explicitly label non-relevant images, so as to achieve a better separation between relevant and non-relevant items (Datta et al., 2008). Hong et al. (2000) implement relevance feedback using a Support Vector Machine (SVM) (Vapnik, 1995). Their paper demonstrates the usefulness of SVMs for the relevance feedback problem, but they highlight the need for multiple positive and negative examples to achieve good accuracy. Tong and Chang (2001) utilize an SVM with active learning. Rather than after each round of relevance feedback presenting to the user the top ranked relevant images from a static classifier, their approach is to present images for which the classifier is most uncertain and after each round the classifier determines new decision boundaries. This approach offered improvement over the use of regular SVMs. Hoi et al. (2004) addresses the class imbalance problem, that is for a typical relevance classification task there are far more non-relevant than relevant images. They build a modified SVM referred to as BSVM (Biased Support Vector Machine) which uses spherical hyperplanes to encompass relevant images. Liu et al. (2008a) also address the class imbalance problem, through utilizing Semi-Supervised Learning (SSL) and dimensionality reduction in a method they term “Relevance Aggregation Projection (RAP)”. The authors refer to the asymmetry problem in subspace CBIR machine learning, which is that images labeled as ‘relevant’ share some semantic properties, whilst ‘non-relevant’ images have no common properties, only that they differ from the relevant images. By performing dimensionality reduction and SSL they are able to capture nearby unlabeled data points to relevant points, thus leveraging the unlabeled data to improve classification accuracy. For clarity, the difference between active learning and SSL, is that in active learning, the user is required to annotate after each iteration, unlabeled data which a classifier is most unsure of, whilst in SSL, unlabeled data is assigned to relevant and non-relevant labels through transduction. The two approaches can be complimentary.

Relevance feedback has also seen use in video retrieval domains. Yan et al. (2003) use pseudo-negative relevance feedback in experiments on the TREC Video 2002 corpus. In this work, positive examples for feedback are items in the TREC topic description, whilst negative examples are sampled from low ranks from an initial search using the positive examples (although later research demonstrates a better sampling strategy for pseudo-negative examples is a random sampling (Natsev et al., 2005)). An SVM is used as the classifier, the results of which are combined with the initial ranking, with the approach showing some improvement over the baseline ranking.

Amir et al. (2005) highlight the problem that for users of CBMIR systems query formulation is a complex process, as often it is expressed as a multimodal query which may incorporate text, visual and semantic concept data. Their work is in query reformulation, such that at the end of either a manual or interactive search session, the user is left with a "well crafted multimodal search query" (Amir et al., 2005). Their approach is heuristic and does not utilize SVM's, rather making use of static weights for feature combination, term updating through relevance feedback, and Boolean operators for negative examples (i.e. use of NOT).

Luan et al. (2008) propose a method for interactive video retrieval which leverages multiple forms of feedback for a search session. After an initial ranking, three types of feedback are available to the expert user, for the novice user a recommendation mechanism exists to suggest the type of feedback to select. The three modes are 'Recall-driven Relevance Feedback' (RRF), 'Precision-driven Active Learning' (PAL) and 'Locality-driven Relevance Feedback' (LRF). RRF is designed to illicit as many relevant annotations from the user as possible, by updating the initial ranking using only text and semantic concepts, resulting in quick query times. PAL is an active learning approach, similar to those previously described, whilst LRF exploits the temporal nature of video by returning along with ranked shots, shots which are temporally adjacent. The intention of this work is that different types of queries for different video domains require different feedback strategies to achieve optimal performance.

5.6.2 Query Performance Prediction

Query Performance Prediction is a related area of work which may have application in the task of weighted data fusion. The objective of query performance prediction is for any given query to a search service, determine how likely it is to provide good results to a user. The typical application for this is to identify queries which may perform poorly and are therefore good candidates for query re-formulation (Cronen-Townsend et al., 2002). The potential application for these approaches is that as they have some discriminating power to determine good from poor performing queries, they may be able to provide a good estimation for values of RC_{ij}.

The *clarity score*, defined by Cronen-Townsend et al. (2002), is one of the first post-retrieval methods to attempt to determine how likely a query is to perform well, where performing well means achieving a good Average Precision (AP) score. The main idea is that once a query has been issued, the top set of documents returned can be analysed to determine the level of 'ambiguity' they contain. Using a language modelling approach, they compare the models of the top ranked documents to that of the collection. If the models are similar, then the results are likely to be poor (i.e. ambiguous) as the result set's documents contain similar distributions of indexed terms to that of the document collection. Alternatively if the models greatly differ, it indicates the top documents are about a single topic and are more likely to produce a good ranking (i.e. would produce a high AP value). This approach was extended by Hauff et al. (2008), termed *Improved Clarity* which incorporated an automatic approach for selecting the number of 'top ranked' documents and improved smoothing.

Zhou and Croft have also been very active in this area, proposing a number of measures for query performance prediction (Zhou and Croft, 2006)(Zhou and Croft, 2007). The first of these, the *robustness score*, takes a document collection C and generates a corrupted collection C'. A query is then issued against both collections, if the two result lists are similar, the proposed inference is that the ranking is 'robust' and therefore the query is good. Alternatively if the result lists are different the query is poor as the result was adversely effected by noise which was not ignored by the ranking algorithm. This approach showed some improvements over the *clarity score*, however it assumes that a good ranking is always possible, which in the case of some experts, such as low-level visual features (Chapter 3) does not always hold. Additional methods proposed by Zhou and Croft include *Weighted Information Gain (WIG)* and *Query Feedback* (Zhou and Croft, 2007). *WIG* is an entropy based approach which computes for the 'top' documents for a given query, how likely they are to be relevant compared against the likelihood of relevance from the average document for a collection. *Query Feedback* is similar to the *robustness score*, it posits that a retrieval system is like a noisy communications channel which transforms a query into a result set. By reversing this and generating a synthetic query from a result set, it is possible to issue this synthetic query to obtain an alternate ranked list. The two ranked lists are then compared and the degree of overlap measured, where high overlap indicates a good query.

These approaches demonstrate some promise, particularly for uses such as query reformulation. However there are drawbacks for their implementation in our CBMIR setting. Firstly many of these approaches appear to have a collection effect impacting upon performance (Hauff et al., 2008), whilst still maintaining a range of variables that are required to be set. Secondly, these approaches typically involve a content-based inspection of a subset of the returned documents for a given query. Whilst this may be possible to implement in a language model based CBMIR system, open questions remain regarding the cross-comparability of query prediction scores across heterogeneous experts. That is, it may be that an additional weighting scheme is required in order to appropriately compare these scores, which brings us back to our initial data fusion problem.

5.7 Conclusions

In this chapter we have reviewed approaches to weighted data fusion for the combination of evidence. We began with an examination of past research into data fusion, and hypotheses as to why data fusion improves the effectivness of retrieval systems. Following this we examined the major approaches which can be utilized for the weighted combination of result sets in our matrix of results $\mathbf{RS}$. The majority of approaches we identified typically only allowed the generation of weights to the level of $\mathbf{RC_i}$, that is only at the expert level or potentially just beyond that. Few approaches allowed us to generate or apply weights at the granularity of individual result sets $rs_{i,j}$. Therefore there exists a gap for the development of approaches which can construct the weighting matrix $\mathbf{RC_{i,j}}$. As demonstrated in the previous chapters, the imposition of a hierarchy to weighting approaches places an artificial ceiling on the potential performance achievable, compared to what could be obtained with weighting of all elements in the system.

Chapter 6

Query-Time Data Fusion Using Score Distributions

In this chapter, we will define our own set of algorithms for creating the weights to be used in linear weighted data fusion. We first elaborate on the motivation for our proposed approach introduced with the outcomes of Chapter 4 and the observation of current techniques in Chapter 5, so as to establish where it fits within the 'family' of weighted data fusion algorithms. Secondly we provide an overview of some aspects of ranked result lists which we aim to exploit. Third we present our algorithms for creating our linear weights, followed by experiments on the corpora we have utilised throughout this thesis. Finally we analyse our results, and attempt to determine why aspects of our algorithms work, where improvement can be found, and how our algorithms relate to existing approaches.

The work we are presenting in this chapter has seen previous application in benchmarking evaluations, with the initial version first described in Wilkins et al. (2006b). We subsequently utilized our approach in the TRECVID 2007 benchmark and achieved the top run for visual only sources of evidence. More recently our approaches were extended by the Chinese Academy of Sciences (CAS) to incorporate semantic concept detection outputs, the resulting algorithms claiming the top runs in automatic search in TRECVID 2008 (Cao et al., 1998). We have also applied the approaches described in this chapter to the task of merging semantic concept detector outputs in order to improve effectiveness (Wilkins et al., 2007b).

6.1 Motivation

In this thesis so far, we have reviewed the approaches which are currently utilised for creating the weights to be used for weighted linear data fusion. We have also conducted what we believe to be a thorough empirical investigation of linear weighted data fusion and in particular we have determined what attributes are key to maximising weighted combination for CBMIR. For clarity we re-iterate our terminology used in this thesis:

$$E = \{expert_1 \; ... \; expert_i\}$$

$$Q = \{query_1 \; ... \; query_j\}$$

$$R = \{document_1 \; ... \; document_m\}$$

$$document_m \mapsto (name, rank, score)$$

$$\mathbf{RS} = [rs_{i,j}]_{|E| \times |Q|}$$

$$\mathbf{RC} = [rc_{i,j}]_{|E| \times |Q|}$$

From the previous chapter we have established several criteria that can be used to evaluate the likely success or capabilities of a weighting scheme for data fusion. Firstly, we identified that the imposition of combination 'levels' such as combining the results of queries from experts into a single result for that retrieval expert, places a cap on the performance that can be obtained as compared to combining individual $rs_{i,j}$ without any intervening aggregation. Secondly, we established that it is the specific weighting of pairs $\langle Expert_i, Query_j \rangle$, in other words $rs_{i,j}$, that contributes to good retrieval performance. This means that a data fusion scheme needs to be able to generate weights at the level of $rc_{i,j}$. Finally, an examination of the ideal weights produced for linear data fusion reveals that these weights approximate a log-normal distribution, that is the majority of weights are clustered around the mean weight whilst specific pairs $\langle Expert_i, Query_j \rangle$ are assigned large weights, resulting in the extended tail seen in a log-normal distribution.

Reviewing the approaches we examined in Chapter 5, the majority of these approaches only allowed weighting at the expert level, that is weights were only created for $\mathbf{RC_i}$, and therefore are unable to create weights for pairs $\langle Expert_i, Query_j \rangle$. The task of creating a complete weighting matrix $\mathbf{RC}$ by necessity becomes a *query time* operation, as we cannot know in advance what multi-modal multi-part query may be submitted to the CBMIR system. Of the approaches examined in Chapter 5, the two approaches which allowed this functionality to occur were the score distribution approaches (see 5.5) and to a lesser extent the query-performance prediction approaches (see 5.6.2). The drawbacks of the query-performance approaches were that the established methods appeared to have a degree of collection dependence (Hauff et al., 2008), whilst requiring some degree of *content* inspection post-retrieval.

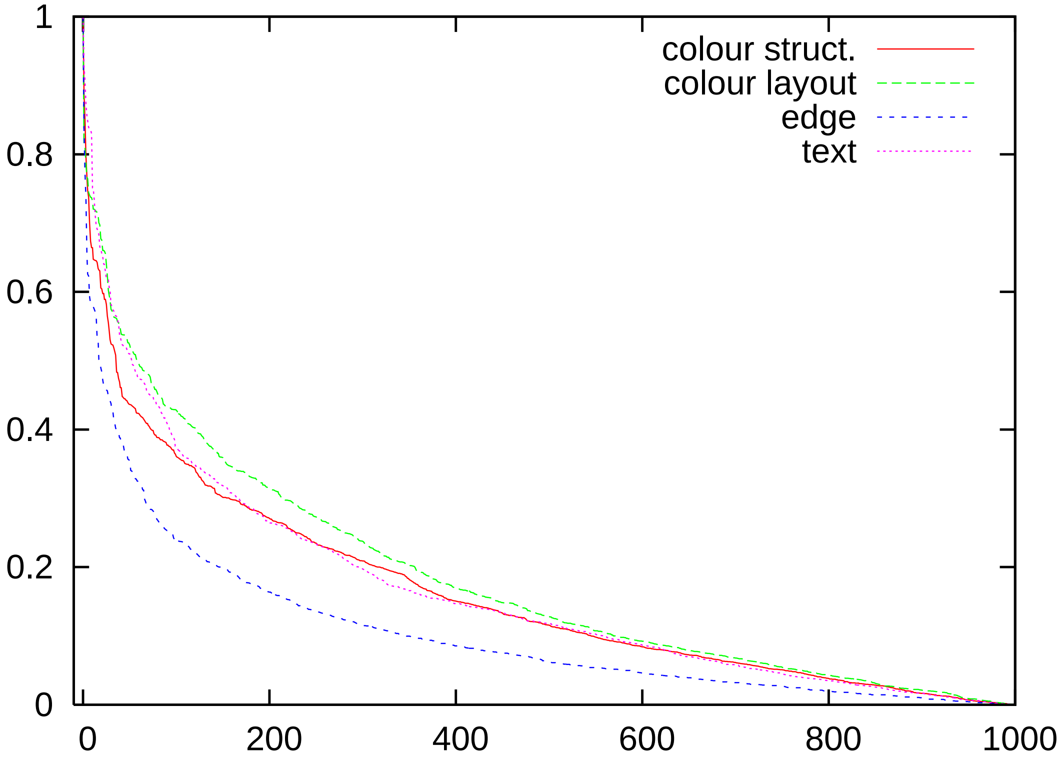
As we have previously covered, the use of score distributions for inferring the degree of relevance has a long history. These various techniques each have attempted to determine if the distribution of relevant documents has some unique properties which allows for the identification of relevant documents from a ranked result set. These techniques have seen various applications, amongst them the creation of weights for data fusion (Manmatha et al., 2001). Robertson (2007) identified various theoretical problems with the current score distributional approaches however, which indicates that there may be issues with applying these approaches to retrieval data fusion problems.

The key characteristic of these approaches were that they were attempting to infer the *absolute* degree of relevance of a given result set. When we are creating weights for result set combination, the weights are not applied in a vacuum, rather the weights are *relative* to what information is being combined. When we have two ranked lists, α and β, and we assign weights respectively of 0.8 and 0.2, we are saying that α is four times more important than β for the purposes of combination. We are not making any absolute judgement through weight assignment of the likelihood of absolute performance. Therefore the objective of any weighting scheme should not be to infer absolute performance, but rather what is the *relative* performance of the elements being combined.

6.2 Score Distribution and Average Precision

Our approach to query-time weight determination is an examination of the score distribution for each result set $rs_{i,j}$ and how it differs from the other $rs_{i,j}$ used for that query, so as to determine the *relative* differences between them. Based upon observation of the variance of score distributions generated for each $rs_{i,j}$ for any given query, we observed that a weak correlation appeared to exist by the rate at which the early *normalised* scores of a $rs_{i,j}$ changed, relative to the other $rs_{i,j}$, and average precision. In other words, a $rs_{i,j}$ whose initial normalised scores changed rapidly was more likely to have a higher AP score than $rs_{i,j}$ which had a more gradual change in document scores. We illustrate this with an example of topic '0135' from TRECVID 2005 in Figure 6.1.

In this graph we see distributions from four $rs_{i,j}$ one each from a text, edge, colour layout and colour structure expert, truncated to the top 1,000 results, with all scores normalised to the range [0..1] using MinMax normalisation. Corresponding to this we can see what AP score each of these achieved. We can see in the table that the 'edge' result scored the highest AP value, and correspondingly in the graph we can see present that it featured the greatest initial change in scores as compared to the other results. For instance, at rank 175, the edge result has a score of approximately 0.2, whilst the colour experts have scores of 0.3, therefore the edge result has undergone a more rapid change in score earlier in the ranking. From this data however, the text result performs well but does not distinguish itself from the other results in the graph presented. In this case, this is because the results are



$rs_{i,j}$	Text	Edge	C. Layout	C. Struct.
AP	0.1611	0.3214	0.0154	0.0032

Figure 6.1: Score Distributions from topic '0135' TRECVID 2005 and AP scores, normalized score on the Y-axis, rank on the X-axis.

truncated to 1000 results. In Figure 6.2 we present the same data, but this time without any result set truncation.

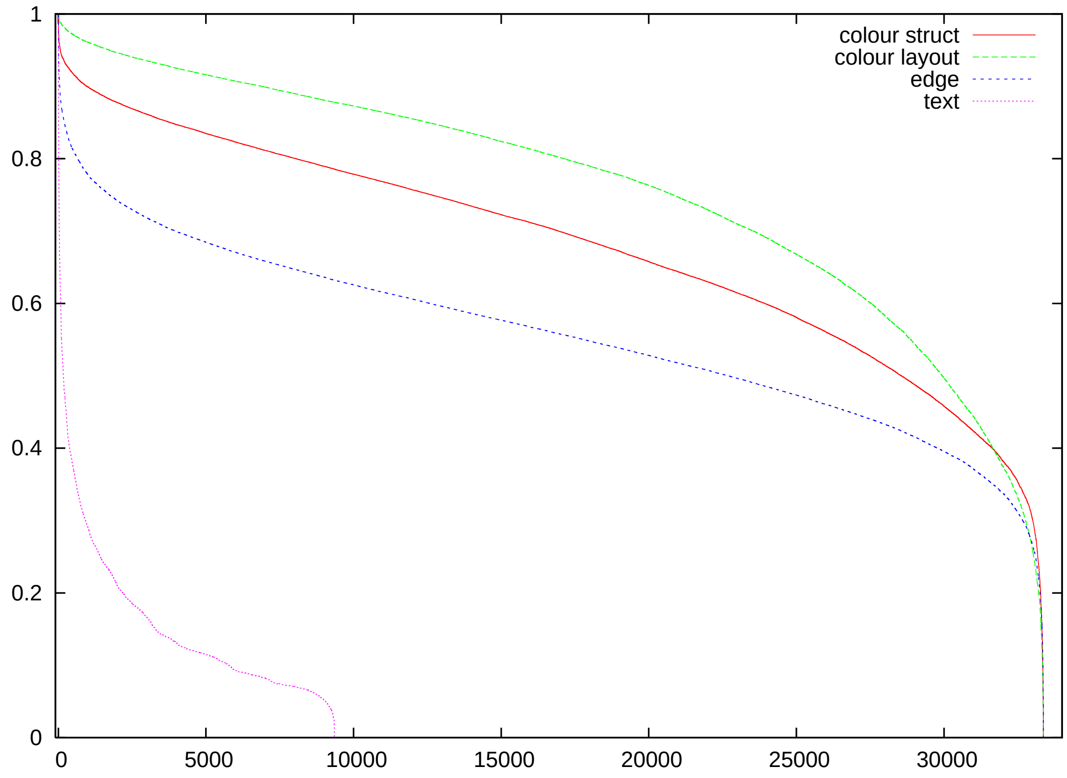


Figure 6.2: Score Distributions from topic ‘0135’ TRECVID 2005, no truncation, normalized score on the Y-axis, rank on the X-axis.

In this view of the data we can see more clearly demonstrated the different rates of score decrease between the result sets examined. From this wider view of the result sets we can see a more clear stratification of the score progression from each of the result sets examined, allowing the text expert to better demonstrate that it too does undergo a more rapid change in score, like the edge expert, in comparison to the two colour experts. This indicates that any comparison of the rate of change of scores between different result sets needs to take into account the relative amounts of the results returned. Nevertheless there appears to be properties of the rate of change of the scores of result sets that weakly correlate with AP such that it is worth investigating if these properties are exploitable.

We are not making the observation that there is a clear, unambiguous correlation between the rate of change of the scores of a result set and AP, however we believe there is some relationship between these two properties. Table 6.1 presents the results of an analysis we conducted of the correlation between the 'area' of the normalised scores of the top 500 results for each $rs_{i,j}$ and AP. Area in this case is determined as the sum of the MinMax normalised scores for the top 500 documents in each $rs_{i,j}$.

TV2003	TV2004	TV2005	TV2006	TV2007	IC2007
-0.264	-0.278	-0.122	-0.052	-0.004	-0.276

Table 6.1: Pearson correlation of area to scoring function and AP.

From this data we can see that a weak correlation is present between the area of the score of the top 500 documents and AP, particularly for corpora TRECVID 2003-2004, ImageCLEF 2007 and to a lesser extent TRECVID 2005. This correlation is quite weak, however it is consistent. If the relationship was an artefact of random noise, we would expect to see at least some of the correlations in the positive range. Furthermore the weakness of the correlations are partially attributable to the dismal performance that individual $rs_{i,j}$ achieve, as we have previously demonstrated and can be seen again in Figure 6.1 with the performance of the colour results. Figure 6.3 presents a scatter-plot of the area of the top 500 documents for each $rs_{i,j}$ and AP for the TRECVID 2003 collection, demonstrating the negative correlation that as the area decreases, the AP score increases. Again of note in this graph is the conflation of multiple topics, each of which varies greatly in performance.

From the data we have presented, we believe we have established a case that a weak correlation exists between the rate of change of the normalised scores of a result set and AP. This correlation however is a *relative* correlation between the result sets used in any given topic, it does not provide any indication of the likely performance of any individual $rs_{i,j}$. What this observation does allow for is the development of methods which can create weights which allow for the relative weighting of result

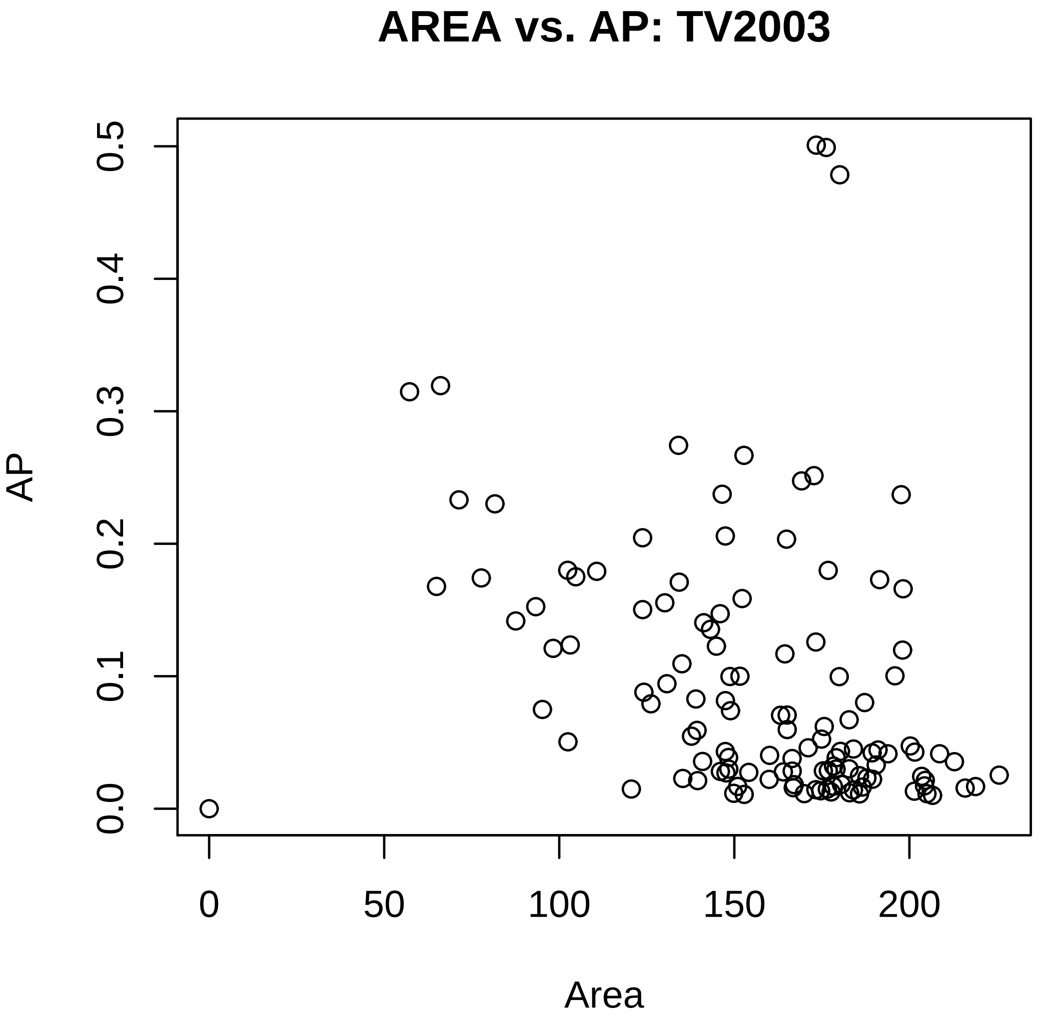


Figure 6.3: Area vs. AP, TRECVID 2003

sets and their likeliness to perform better than the other result sets used for that particular query. In the next section we will present our methods for attempting to leverage these observations.

6.3 Algorithms

As stated in our introduction, we have developed approaches which leverage these relative changes in score distributions to create methods which generate weights for linear data fusion. The key characteristics of these approaches are that the weight generation occurs at *query-time*, which therefore allows for a complete populating of the weighting matrix **RC**. As such, the algorithms we have defined are *unsupervised* and are not reliant on training data being available. These two key characteristics distinguish our approaches from current approaches to weight generation. In this section we define two approaches, known as Mean Average Distance (MAD) and Maximum Deviation Method (MDM). The first approach, MAD, is the approach our previously published research is based upon (Wilkins et al., 2006b), whilst MDM is our recent development which offers performance improvement.

6.3.1 Mean Average Distance (MAD)

Our model for relative weight determination is based upon an examination of the differences between the scores of adjacent documents in one list and contrasting that to the differences between the scores of adjacent documents in another result list. We refer to this as the Mean Average Distance (MAD), and formally this is given by:

$$MAD = \frac{\sum_{n=1}^{N}(score(n) - score(n+1))}{N-1} \tag{6.1}$$

Where *score* is the MinMax normalised document score, N is the total number of documents to be examined in the result set. For example, if document ‘A’ has a normalised score of 0.85 and document 'B' has a normalised score of 0.80, then the difference we measure between the two is 0.05. We can then sum these differences for each result set to provide an indication as to the progression of the document scores for a result set.

A direct comparison of these differences in itself will not yield much useful information as the differences could be accounted for by the ranking metric or even the nature of the distribution of the raw feature data. Therefore to achieve a score that is comparable between lists we define a ratio which measures MAD within a top subset of a result list, versus that of a larger set of the same result list. The resulting score we refer to as a Similarity Cluster (SC), and can be defined as:

$$SC = \frac{MAD(subset)}{MAD(largerset)} \tag{6.2}$$

In effect the variables *subset* and *largerset* are substituted into Eq 6.1 for the variable N. The selection of the values of *subset* and *largerset* dictates how aggressively we weight the change in document scores for a $rs_{i,j}$ which occur earlier in the ranking. Whilst $rs_{i,j}$ from visual experts will always retrieve the same number of documents for a given query, as visual expert rankings are based on a similarity measure, results set sizes from a text expert will vary in size of the number of non-zero scored documents returned, highlighted in our earlier examination of score distributions for topic '0135'. Therefore we assign the values of *subset* and *largerset* as percentages of the result set size, which through testing we have determined to be 5% and 95% respectively. This means that we compare the change in score of the top 5% of a result set, versus the change in score of 95% of the result set. This ratio provides us with a measure of how much the initial scores of a result set change in comparison to the overall change in scores for a result set. If the average change of the subset is greater than the overall average change, a large SC value will be generated, and vice versa for a small SC value. The final weight for each $rc_{i,j}$ is the scaling of the SC values to the range [0..1] where $\sum rc_{i,j} = 1$, as given by Eq 6.3.

$$rc_{i,j} = \frac{rs_{i,j}\ SC\ Score}{\Sigma\mathbf{RS}\ SC\ Scores} \tag{6.3}$$

6.3.2 Maximum Deviation Method (MDM)

The maximum deviation method (MDM) is an evolution of the previously detailed MAD algorithm both of which share the same intent, to create greater weights for those $rs_{i,j}$ whose scores undergo a more rapid initial change in score as compared to the other $rs_{i,j}$ used for that particular query. Whilst the previous method examined the average change in scores for given sizes of a result set and compared these changes, the MDM algorithm instead uses a fixed reference point from which to compare all result sets.

From Section 6.2 we observed that a rapid change in score was an indicator of potential better AP performance as compared to the other results sets used. Therefore the converse of this position is that a gradual change in document scores is an indicator of likely poor relative performance. An example of a poor score distribution would be a linear progression from the maximum to the minimum score, which we refer to as a linear ranking.

If we compare the normalised scores of a $rs_{i,j}$ to that of the normalised scores of a linear ranking we can determine two key variables. The first is that we can compare for each rank the distance between the actual rank's score and the score of the linear rank, which we refer to as d. By comparing both distributions we can determine what the maximum value of d is, which would correspond to the point which maximally deviates from a linear ranking. The second variable is the rank position at which the maximum deviation occurs, which we refer to as r. We can illustrate both of these variables in Figure 6.4, which compares a $rs_{i,j}$ score distribution, plotted as the green line, against the reference linear distribution, plotted as the dashed black line. We can more formally define the variables d and r in Equations 6.46.5.

$$d = max\ (\ linear(x) - current(x)\ ),\ for\ x \in X \tag{6.4}$$

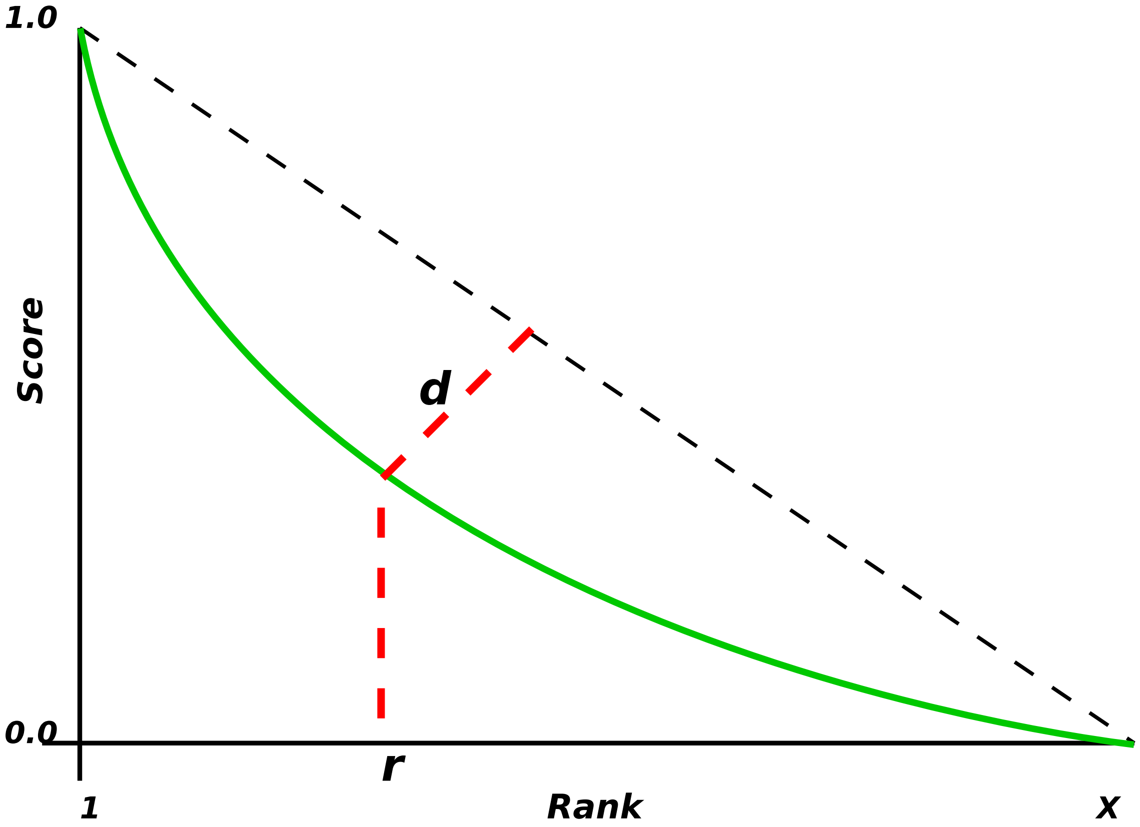


Figure 6.4: MDM Algorithm Visualisation

$$r = \frac{rank(d)}{|X|} \tag{6.5}$$

where max is a function which returns the maximum value of the comparison, x is the rank position being examined from the set of all ranks X for any given $rs_{i,j}$. The functions $linear(x)$ and $current(x)$ return the scores at rank position x for the linear and $rs_{i,j}$ distributions respectively. The function $rank(d)$ returns the value of x where the maximum deviation occurred, and $|X|$ is the size of $rs_{i,j}$. One caveat for these formulae is if the difference between the linear score and the current score is negative, that is that the score is actually more than the linear score. In cases where this occurs we set the difference to 0. If for a given $rs_{i,j}$ the maximum difference is 0, that means the actual score distribution was above linear for all ranks x, therefore we assign an arbitrary low weight of $\frac{1}{1000}$ for that $rs_{i,j}$. The final calculation of the weight for each $rs_{i,j}$ is given in Equation 6.6. Like the previous MAD method, we also scale these values to the range [0..1].

$$MDM(rs_{i,j}) = \frac{d}{r} \tag{6.6}$$

We believe the MDM method better captures the differences between score distributions where the criteria for comparison is the initial change in scores. By comparing against the fixed linear distribution we can determine not only how great a score distribution deviates from the linear distribution, but where this deviation occurs. For instance, if the deviation is greatest towards the end of the ranking, then a low weight will be generated, as the value for r will be high. Likewise, if two $rs_{i,j}$ demonstrate the greatest deviation at approximately the same location, but one instance has a greater deviation, then it will receive the greater weight. This method also improves upon our previous MAD method as it is parameter free.

Both of these methods we have defined have the capability of fully populating the weighting matrix **RC**, rather than just setting expert level weights. A consequence of this is that these approaches are unsupervised, as the weight generation occurs at query-time and is query-dependent. Our selection of direct-level combination for this approach is because the motivation for the development of these algorithms is that they can create weights at the direct-level of combination, an attribute current weighting schemes do not possess.

6.4 Experimental Results

In this section we will be present the experimental results of applying our query-time weight generation algorithms to the data sets we have been exploring throughout this thesis. For these experiments we will conduct both rank and score-based variants of these experiments. Score-based combination will use MinMax for normalisation, whilst for rank-based normalisation we will use BordaMAX. The combination of result sets will be through weighted CombSUM.

Our previously published experiments utilising the MAD approach had employed combination levels in that investigation (Wilkins et al., 2006b)(Wilkins et al., 2007b).

In this series of experiments we will perform the direct level of combination. This is a more challenging environment in which to operate, as the raw number of result sets to be combined is quite high. Table 6.2 demonstrates for each corpora when using a direct level of combination what the average number of result sets to be combined is. All available retrieval experts will be utilised for this experiment, our six visual and one text expert. As such we consider this a more stern test of the capability of our approaches to generate applicable weights, particularly as direct level combination is the position we advocated earlier in this thesis.

TV2003	TV2004	TV2005	TV2006	TV2007	IC2007
32.88	37.25	57.75	43.25	52.25	19

Table 6.2: Average number of $rs_{i,j}$ to be combined per query, per corpora.

We present our results in Figures 6.5 and 6.6, first the rank-based results, then the score-based results. Our comparison for this experiment is the uniform combination of result sets to demonstrate the effect of no weighting. For each result we compare MAP, recall and P10. Runs whose MAP is in bold are significantly different to the baseline run. To demonstrate the difference in performance between our query-dependent approaches and the baseline, we present in Figure 6.7 the percentage improvement for each approach over their respective baselines for MAP, and in Figure 6.8 the improvement when P10 is examined.

The presented results demonstrate many aspects of performance. Firstly in a majority of cases, our query-dependent approaches are able to create weighting matrices **RC** which improve performance over the baseline results. The one notable exception to this is the TRECVID 2007 corpus, which would appear to be somewhat of an outlier. Secondly we observe that in these cases, score based normalisation clearly outperforms rank-based normalisation, contrary to our earlier findings when the ideal weighting set was employed. Again there is one exception to this, which

Legend	TRECVID 2003			TRECVID 2004		
Run	**MAP**	**Recall**	**P10**	**MAP**	**Recall**	**P10**
Uniform Rank	0.0506	0.2815	0.0880	0.0280	0.1673	0.0957
MAD Rank	0.0544	0.2738	0.0960	0.0306	0.1729	0.1043
MDM Rank	0.0638	0.2704	0.1080	0.0327	0.1833	0.1261

Legend	TRECVID 2005			TRECVID 2006		
Run	**MAP**	**Recall**	**P10**	**MAP**	**Recall**	**P10**
Uniform Rank	0.0602	0.2215	0.2208	0.0189	0.0984	0.0958
MAD Rank	**0.0619**	0.2205	0.2458	0.0197	0.1020	0.1083
MDM Rank	**0.0632**	0.2164	0.2375	0.0201	0.1143	0.1083

Legend	TRECVID 2007			ImageCLEF 2007		
Run	**MAP**	**Recall**	**P10**	**MAP**	**Recall**	**P10**
Uniform Rank	0.0445	0.2700	0.1833	0.1269	0.4950	0.3183
MAD Rank	0.0427	0.2640	0.2000	0.1360	0.5318	0.3150
MDM Rank	0.0368	0.2545	0.1917	0.1423	0.5473	0.2967

Figure 6.5: Rank based query-dependent weighting

is the TRECVID 2006 corpus, which throughout our experiments has consistently been the worst performing of our six corpora. In TRECVID 2006 the rank-based normalisation approaches outperform the score-based approached. TRECVID 2006 as the poorest performing corpora can also be considered the most noisy, which indicates that the extreme smoothing offered by ranking is of greater benefit (Croft, 2000).

It is of interest however the discrepancy between the benefits of using ranks with optimal weights, as in our previous experimentation chapter, versus using scores with sub-optimal weights. Clearly appropriate weights are the key to obtaining strong performance for noisy data fusion tasks such as CBMIR. From the presented data, we observe that our greatest performance improvements with our query-dependent weighting approaches came when ranks were utilised (see Figure 6.7), whilst these relative improvements were less when score-based normalisation was utilised. Conversely however in absolute performance the score-based methods out-performed the rank-based methods. An examination of the uniform runs demonstrates that score-based combination typically outperforms rank-based combination when no weighting

Legend	TRECVID 2003			TRECVID 2004		
Run	**MAP**	**Recall**	**P10**	**MAP**	**Recall**	**P10**
Uniform Score	0.0618	0.2719	0.1080	0.0296	0.1708	0.1000
MAD Score	0.0640	0.2715	0.1160	0.0316	0.1746	0.1043
MDM Score	0.0672	0.2688	0.1200	0.0336	0.1837	0.1217

Legend	TRECVID 2005			TRECVID 2006		
Run	**MAP**	**Recall**	**P10**	**MAP**	**Recall**	**P10**
Uniform Score	0.0643	0.2172	0.2250	0.0177	0.0972	0.0792
MAD Score	**0.0669**	0.2191	0.2458	0.0186	0.1009	0.1042
MDM Score	**0.0678**	0.2204	0.2542	0.0172	0.1040	0.1083

Legend	TRECVID 2007			ImageCLEF 2007		
Run	**MAP**	**Recall**	**P10**	**MAP**	**Recall**	**P10**
Uniform Score	0.0532	0.2567	0.1750	0.1420	0.4922	0.3700
MAD Score	**0.0552**	0.2605	0.1792	**0.1558**	0.4881	0.3867
MDM Score	0.0509	0.2545	0.1792	0.1636	0.5148	0.3767

Figure 6.6: Score based query-dependent weighting

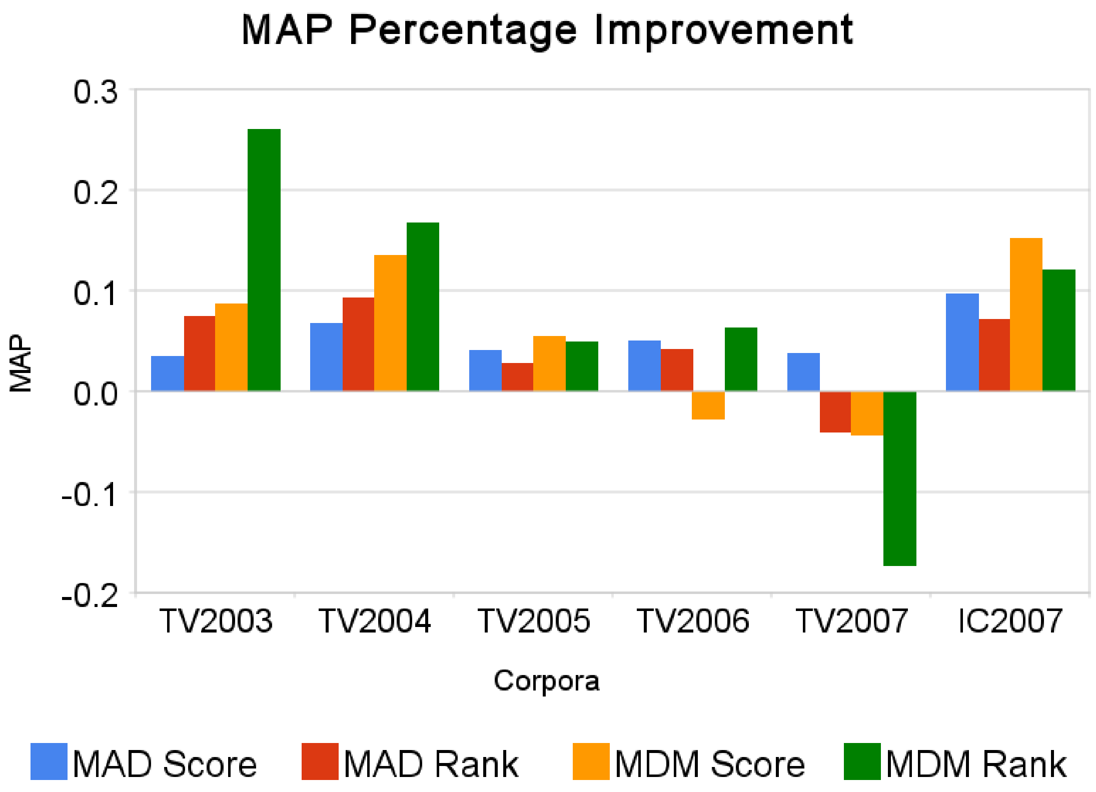


Figure 6.7: MAP Improvement over baseline

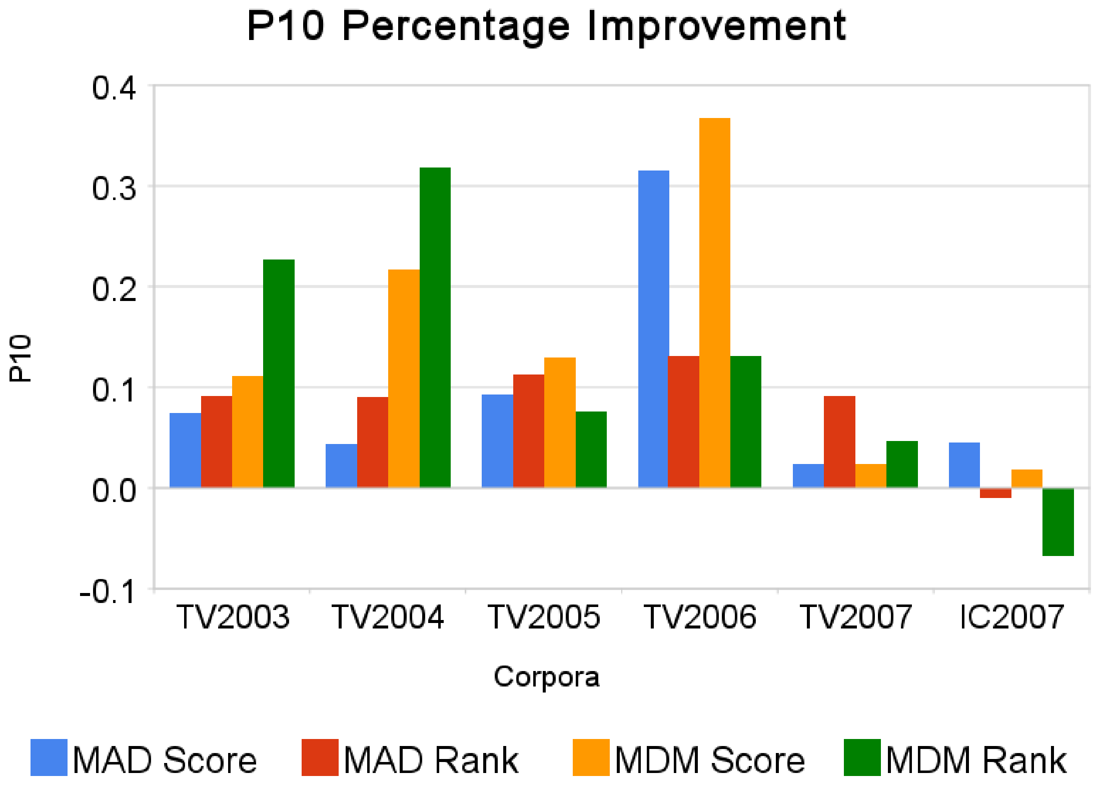


Figure 6.8: P10 Improvement over baseline

is employed. Therefore we have observed that if the weights are optimal, rank-based normalisation produces the greatest performance, as the weights purely dictate performance. If the weights are sub-optimal, score-based normalisation provides a level of performance improvement as the scores regulate the impact of the weights. That is, as the scores themselves are non-linear, they provide an indication of which documents are more important than others, thus the application of the weight to a score has less impact than the application of the weight to a rank.

Comparing our two approaches MAD and MDM, we can see demonstrated that the MDM outperforms the MAD approaches in a majority of cases. Again the notable exception to this is the TRECVID 2007 corpora which appears to behave quite differently to each of our other corpora under examination. Both approaches offered improvement for P10 over the baseline, demonstrating that the weights were successfully promoting more relevant documents higher in the ranked list, however at times this was at the expense of recall. Overall however the MDM approach appears to more successfully exploit differences in score distributions than the MAD approach. We can demonstrate this by comparing the correlation of the improvement each approach obtained, versus the data in Table 6.1 which demonstrated which corpora had higher correlations of ‘area’ and AP. The results of this correlation are presented in Table 6.3.

	Rank	Score
MAD	-0.88	-0.55
MDM	-0.96	-0.85

Table 6.3: Correlation of MAP percentage improvement and area correlations from Table 6.1

This table demonstrates a very strong correlation between the improvement achieved over the baseline result, and the degree to which a given corpus displayed a correlation between the area under a score distribution and AP. We can see that for both MDM approaches, a strong correlation is present, indicating that performance is dictated by this underlying association of area and AP (see Section 6.2). What is of particular interest, is that TRECVID 2003, 2004 and ImageCLEF 2007 displayed the highest levels of correlation between area and AP, and that these corpora were the best performing with regards to improvement over the baseline approach. This is useful as it points towards the development of collection-based tests which may be employed to determine the benefits of using our query-dependent weighting approach within that domain.

The results when examined for statistical significance are disappointing, as only one run in the rank-based evaluation demonstrates a significant difference, whilst for the score-based runs only three runs demonstrate a difference. We believe however that this artefact is due partly to the nature of the significance test, coupled with nature of TRECVID topics. The significance test as indicated previously tests for significance between pairs by randomly flipping several topic results of the two lists. The issue as we have highlighted previously, with respect to MAP and TRECVID, is that in several TRECVID evaluations performance is dominated by a handful of very high performing topics. This is effect can mask performance differences in the aforementioned statistical test. For several of the evaluations if we remove one of the high performing topics, whilst we see a degradation in overall MAP, the significance test reports significant differences between the runs. However the selective removal of topics from evaluation benchmarks is an unwise practise, so it will not be further pursued. Nevertheless an examination of the results, particularly the percentage increase graphs demonstrates a deterministic process as the ordering and magnitude of the differences remains constant across corpora. Specifically the MDM runs consistently out-perform the MAD runs across all corpora, with the exception of TRECVID 2007 where the magnitude of the decrease in performance is also consistent.

6.5 Analysis

From the results from our experiment we have demonstrated that it is possible to leverage the relative difference of score distributions in order to create query-time weights for data fusion. Whilst there's no doubt that the methods we have presented can be improved upon, a more fundamental question is why do these approaches achieve any success at all? Firstly however, we need to examine the output of the two algorithms we have created, namely what was the final distribution of weights generated by the two approaches. Using the methods we utilised in Chapter 4, we present for both MAD and MDM the histograms of the weights generated, and an analysis of their distribution through Q-Q plots against normal distribution. These graphs are given for the MAD approach in Figure 6.9 and MDM in Figure 6.10

Examining these graphs we can see demonstrated that they do in fact generate quite different sets of weights, despite having the same motivation for their implementation. The weights generated by the MAD approach approximate a normal distribution, which is evidenced by both the histogram, and the tracking along the dashed line in the Q-Q plot. Encouragingly, the MDM approach, which we would consider the more successful of the two approaches, generates a weight distribution which is far closer in shape to what the ideal weight distribution shape is. The MDM approach approximates a log-normal distribution, which we can see in the histogram and on the Q-Q plot. Compared to the ideal weight distributions which we examined in the previous Chapter, this log-normal is more compact, however the general form of the weights it is generating is very encouraging. From the performance figures there is clearly room for improvement, however the indications are that this method is proceeding in a promising direction.

The fundamental issue as mentioned in the beginning of this section is why this attribute occurs at all. We have developed a working hypothesis of what we believe to be the cause. The benchmarks for multimedia data are typically fairly sparse with relevant data (Natsev et al., 2005), therefore we believe that when a

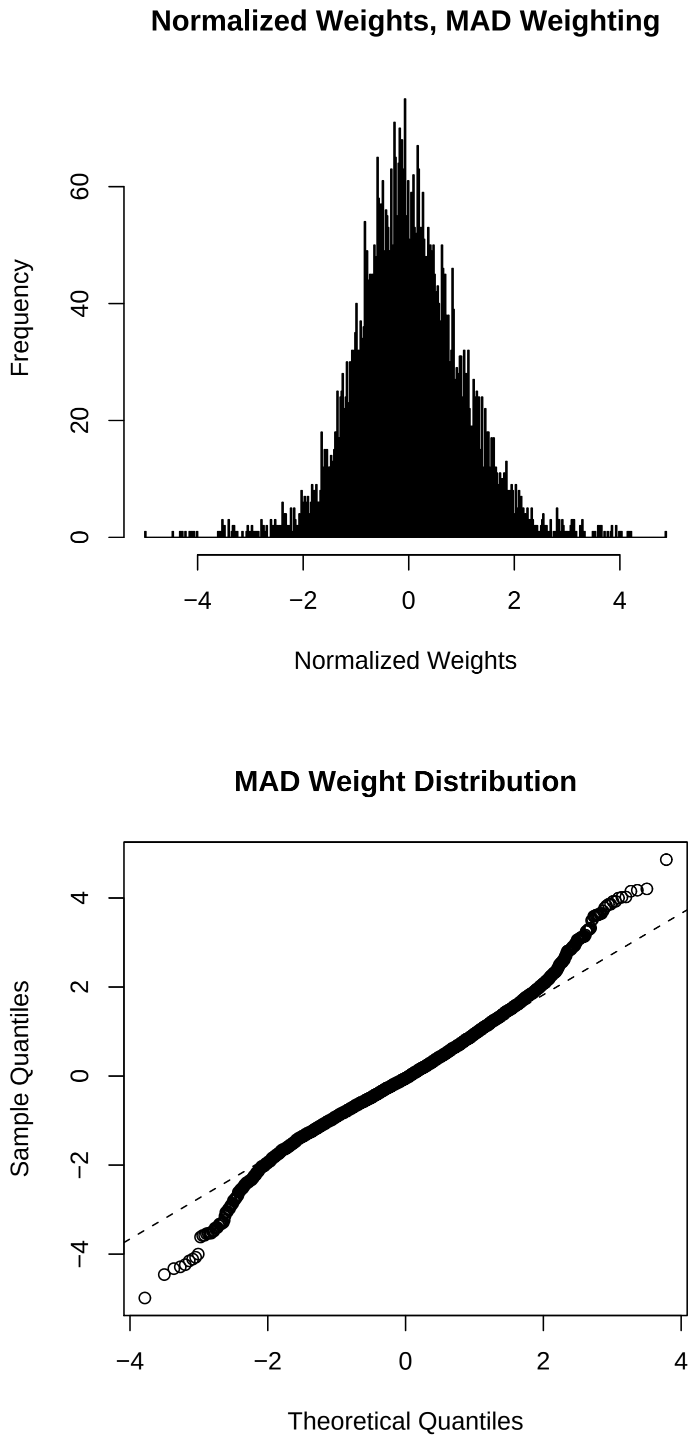


Figure 6.9: MAD Weight Distributions, all corpora

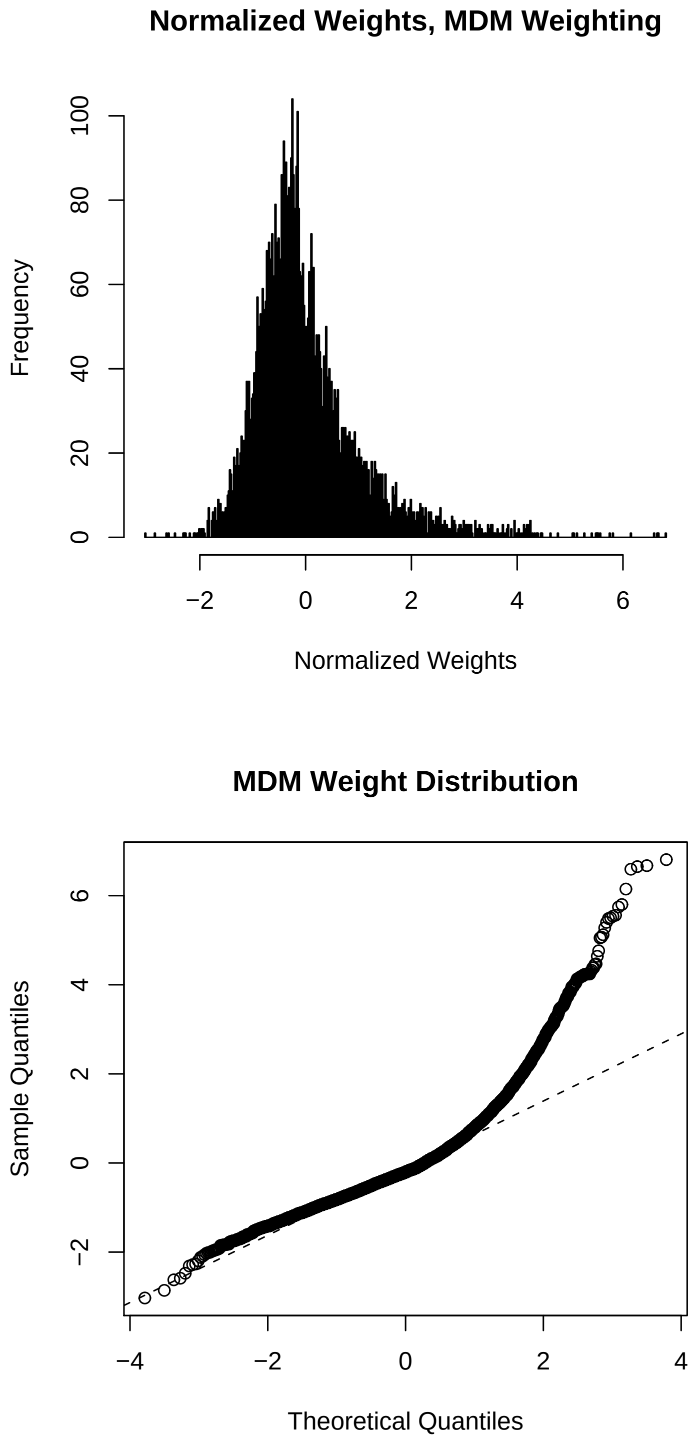


Figure 6.10: MDM Weight Distributions, all corpora

result set undergoes a rapid change in its initial normalised scores, that the expert which is generating these results is making more definitive judgements, or has greater confidence, in the rank positions it is assigning those documents. Conversely, if there is a gradual decline in the normalised scores, then the expert is ranking multiple documents in similar rank positions and is making less determined decisions about the rank position. Both of these viewpoints are not absolute but are made when observing the set of score distributions to be combined, as such these are relative inferences.

Our justification for this hypothesis is based upon the work of Robertson and his investigations into score distributions and probabilistic ranking (Robertson, 1977, 2007). Robertson highlights that the only purpose of a score generated by the system is to allow the system to assign a rank to a document (Robertson, 2007). Robertson has also previously defined the *Probabilistic Ranking Principle (PRP)* which is the assumed link between relevance and ranking, that a system will rank documents in order of their probability of being relevant. Finally we know that as the output of a search system is ranked, its scoring function must be monotonic, that is as the ranking is produced by the ordered scores, we know that when plotted the score distribution will never have a positive inclination.

Given the above, the worst case for any retrieval system therefore is for documents to be assigned the same score, which will produce ties in the ranking. When plotted, the score distribution which contains tied documents will produce a horizontal line. This horizontal line effectively means that for the length of that line, the ordering of the rank positions which it occupies is effectively random. This is the worst possible scenario for a search engine as it has been unable to distinguish any differences between the documents with regards to relevance, yet has produced a ranking which assigns an importance to a document which it does not possess. Therefore we can state that the worst case for a score distribution is a horizontal ranking.

Conversely we can take the position that the best case by extension, is that in which the scoring function assigns unique values to documents such that they are unambiguously ordered. A rapid change in score represents documents of a ranked list which are likely to have the greatest differentiation from the other documents ranked. In an ideal ranking where documents are ordered by their degree or likelihood of relevance, we would expect documents with greater degrees of relevance to appear earlier in the ranking, with the change in score representing this. Equally we would like all non-relevant documents to be assigned the same score, as they are all equally non-relevant. The assignment of uniform scores to non-relevant documents would produce a horizontal ranking, whilst as there are small amounts of relevant documents we would expect a steep gradient representing the progression from very relevant through to non-relevant documents. These properties of score distributions which we have identified represent our best approximation as to the causes of the observed effect and provide us with a justification for its exploration.

6.6 Conclusions

In this chapter we have presented our own approach to weighted data fusion which is capable of generating the complete weighting matrix **RC** which few other data fusion algorithms are capable of. These approaches are query-dependent, but as a consequence is unsupervised. We have demonstrated reasonable performance improvements utilising our algorithms which point to its potential for enhancing retrieval performance.

The task in which we chose to evaluate was weighted data fusion where we employed no weighting hierarchy. This task is quite difficult, as we are generating in some cases over 55 weights for combining sources of evidence. Nevertheless we chose this evaluation as our previous investigation has demonstrated that direct levels of combination are capable of offering superior performance.

Whilst we have advocated our data fusion algorithms, their application does not need to occur in isolation. The algorithms we have defined we believe could quite easily be incorporated into existing data fusion algorithms such as query-class classification. For instance, once the query classes have been established through training, our methods could be utilised at query-time to modify the query class weights such that they suit the specific query being issued. Furthermore our approaches are computable with other retrieval approaches for improving performance, such as relevance feedback, as they can be utilised to improve the initial ranking before feedback.

Chapter 7

Conclusions

Multimedia retrieval, to borrow a cliché from World Wide Web research (WWW), is growing at an exponential rate, with the continued adoption of digital capture devices ensuring an ever-increasing stream of data being generated. Just like search evolved on the WWW with the move to content-based searching, search within multimedia documents is embracing content-based methods in order not just to cope with the deluge of new data, but also to provide relevant results to new search vectors which were not previously available. The development of low-level features has allowed the unsupervised extraction of searchable data from multimedia documents, however the use of just one low-level feature is unsatisfactory as such features generally perform poorly in isolation. Therefore, we must combine multiple low-level feature results in order to obtain acceptable retrieval performance, and this is often achieved though weighted data fusion. This thesis has been concerned with an investigation into the task of weighted data fusion of noisy sources of information in order to improve retrieval effectiveness.

In Chapter 2 we presented a high level overview of Multimedia Information Retrieval (MIR), specifically of Content-Based Multimedia Information Retrieval (CBMIR). We presented the *sensory gap* and the *semantic gap* to highlight the noisy properties of multimedia data which makes the task of retrieval for CBMIR difficult. Whilst we highlighted approaches such as *semantic concept detection* as a means for attempting to bridge, or mitigate, the effect of the semantic gap, we demonstrated the significant challenges that affect that research field. We presented our motivation for focusing our attention on the use of low-level features, otherwise known as low-level *retrieval experts*. Having shown that these experts produce noisy evidence, it demonstrated the need for the employment of effective weighting schemes to appropriately combine these forms of evidence. The combination of these forms of evidence is far from straightforward, however, with many techniques available for achieving this aim.

The objective in Chapter 3 was to identify all explicit and implicit variables which impact upon the performance of the combination of multiple ranked results. The mechanism for this combination is *data fusion*, and we thus provided an introduction to this topic along with an overview of previous general data fusion research. Whilst there have been numerous prior studies examining various aspects of data fusion performance, there have always been variables which are implicitly set and not examined, (such as the use of combination levels) which impact upon performance and required exploration. The factors we identified include normalisation approaches, combination operators such as CombSUM and CombMNZ, hierarchical combination levels and read-depths. In this Chapter we also introduced terminology to help define these factors and their interactions, highlighting that our investigation and subsequent system would be processing multi-part multi-expert queries. Having multiple $Expert_i$ and $Query_j$ available meant that at query time a matrix of results was generated to be combined, **RS** which if each element was weighted would require the weighting matrix **RC**.

In chapter 4, we presented an empirical evaluation of the variables identified in chapter 3 in terms of their impact on weighted data fusion. We introduced an alternative experimental model to that commonly used in the evaluation of IR systems, directly optimising on the test collections so as to find the ideal sets of weights for data fusion. This allowed us to neutralise the impact of the weights on performance so that we could evaluate the actual impact of the other variables, which may otherwise have been masked by the performance of the weights. As a result we made several fundamental observations which challenged accepted wisdom about the parameters to be used for weighted data fusion tasks. These observations included the following: the ideal weights for data fusion approximate a log-normal distribution, when ideal weighting is used rank normalisation outperforms score normalisation, CombSUM clearly outperforms CombMNZ contrary to accepted wisdom and the imposition of combinatorial hierarchies places quite low ceilings on the maximum performance that can be obtained. This chapter also revisited earlier data fusion experiments to test these observations on previously published data sets.

In Chapter 5 we presented a review of approaches which can be used for weighted data fusion. The majority of these approaches typically only allow the generation of weights to the level of $\mathbf{RC_i}$, that is only at the expert level. Few existing approaches allowed us to generate or apply weights at the granularity of individual result sets $rs_{i,j}$. In chapter 4, we demonstrated a need for the development of algorithms for the generation of weighting schemes capable of producing the complete weighting matrix **RC**. This class of algorithm is difficult to develop however, primarily because such an approach would need to generate *query dependent* weights. Query-dependent approaches by definition can only make limited use of training data, which can be an issue when heterogeneous document collections are utilised, such as those we have experimented with.

Finally in Chapter 6 we presented our novel approach to weighted data fusion which comprises an unsupervised *query-dependent* set of algorithms capable of generating the complete weighting matrix **RC**. Our approach demonstrates a reasonable performance improvement, leveraging properties of the score distributions of the result sets to be combined in order to create weights which aid retrieval performance. We conducted an evaluation using a direct level of combination where in some testing corpora we were on average creating 55 weights per query. Whilst showing promise, we highlighted that this approach could easily be incorporated into other data fusion schemes so as to provide them with a degree of query-dependent weighting.

In this thesis we have focused on an examination of the use of low-level retrieval experts for weighted combination within the task of CBMIR. Low-level experts are often derided or overlooked as sources of retrieval effectiveness, with much research today focused on areas such as semantic concept detection as the panacea to the relative poor performance of multimedia retrieval (Smeulders et al., 2000)(Hauptmann and Christel, 2004)(Blanken et al., 2007). This position is understandable,: as we saw in Chapter 3, many individual experts across corpora averaged only 0.01 - 0.03, barely above the level of random noise. Despite this, data fusion is the key process for leveraging these sources of data in order to improve retrieval performance, particularly weighted data fusion. The combination of these forms of evidence through uniform weighting was surprising in the level of retrieval performance that was obtained,: simply fusing these sources of evidence together typically saw an order of magnitude improvement over the performance of individual experts.

Our testing methodology of optimising directly on the test collection such that the ideal weights are used was justified when considered against the empirical observations which we made. The creation of weights was performed within a vacuum in many previous cases as a thorough grid search of the parameter space is infeasible at a topic dependant level. This process meant that the weights used would impact upon the experimental observations. By having the weights at close to their ideal values, the weights themselves became a fixed constant which allowed for the accurate measurement and observation of other factors which impacted upon retrieval performance. This leads to two of our key findings: the demonstration of the effectiveness of CombSUM over CombMNZ contrary to accepted wisdom, and the very large performance cap created by utilising combinatorial hierarchies.

In this thesis, we have conclusively demonstrated that that low-level experts, when weighted with ideal weights and using the appropriate combination operators, achieves a level of performance that runs totally against previous experimental knowledge. The application of the ideal weights sees an order of magnitude improvement in retrieval effectiveness as measured by MAP. This result provides significant impetus for the continued development of approaches for generating weights for data fusion using low-level features, as the performance gains of the correct weighting exceed expectations. This task is certainly non-trivial and as likely to be solved in the near term as finding a solution for determining relevance with complete accuracy. Nevertheless, our research has shown that *many* noisy signals when used in unison are far greater than the sum of their parts, regardless of weighting, and that using the correct data fusion variables and an ideal set of weights retrieval performance is far in excess of all expectations.

7.1 Future Work

There are multiple avenues for work to progress on from this thesis. However there are two key related outcomes which we would like to see receive wider attention, domain specalization, and the observation on the distribution of weights approximating a heavily skewed log-normal distribution.

Firstly, our work in this thesis has been at a generic level, where we experimented on six different corpora so as to get generalised results. However in an operational system, the corpus may be more well known. We would be interested in examining if the techniques we have developed so far can have extension to the corpus level. This would allow for a degree of specialization to occur and enhance retrieval performance. In a similar vein, we treated the retrieval experts we used as equal, making no prior assumptions about their behaviour. An extension we would be keen to explore is if any of these techniques could be used to categorize or tailor the retrieval techniques to specific experts.

Second, the observation that a minority of pairs $\langle Expert_i, Query_j \rangle$ drives performance, readily dictates that we should develop further improvements to our query-time algorithms to exploit this property. Our initial attempts whilst achieving some success can clearly be improved upon when compared against what performance could be attained. This direction of research is challenging and will likely need to incorporate some form of specific content-analysis step, rather than the generic approaches which we have utilised in this thesis. Nevertheless the degree to which performance can be improved overall for CBMIR makes this task a worthy objective.

7.2 Publications

Listed below are some of the major publications that were either worked on, or contributed to, this thesis during my tenure as a PhD student at Dublin City University. As mentioned in my acknowledgements, I am very grateful and lucky to have completed these studies within the Centre for Digital Video Processing (CDVP), which has allowed me to work on and explore a wide variety of research areas.

Publications which directly deal with the algorithms detailed in Chapter 6 include (Wilkins et al., 2006a)(Wilkins et al., 2006b), the application of our algorithms to classifier combination (Wilkins et al., 2007b), whilst also being featured in the SIGIR doctoral consortium 2007 (Wilkins, 2007).

Whilst the majority of this work has been concerned with laboratory style experiments, we have also examined the role of users and their interactions with CBMIR systems (Byrne et al., 2008)(Wilkins et al., 2009). The study of users and the use of CBMIR systems is a thesis in itself and throws up many more challenges than we have covered here, however given the variety of methods in which a user may interact with a CBMIR system we are finding that user variability plays a very large part in retrieval performance.

Throughout the years 2005-2008 the algorithms developed as part of this thesis found testing grounds in the benchmarking activities of TRECVID and ImageCLEF-Photo for these years. In the proceedings of these workshops are papers from Dublin City University (DCU) which will highlight what we developed for that year, and show the evolution of our thoughts, notably (Järvelin et al., 2007)(Wilkins et al., 2007a). Our recent journal (Smeaton et al., 2008) written with several other groups who participate in TRECVID, provides an overview of the types and variations of retrieval systems which TRECVID participants create.

Bibliography

Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying Support Vector Machines to Imbalanced Datasets. In *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*, pages 39–50, Pisa, Italy.

Amir, A., Berg, M., and Permuter, H. (2005). Mutual relevance feedback for multimodal query formulation in video retrieval. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR 2005)*, pages 17–24, Singapore, Singapore.

Amir, A., O Argillander, J., Berg, M., Chang, S.-F., Franz, M., Hsu, W., Iyengar, G., R Kender, J., Kennedy, L., Lin, C.-Y., Naphade, M., (Paul) Natsev, A., Smith, J. R., Tesic, J., Wu, G., Yan, R., and Zhang, D. (2004). IBM Research TRECVID-2004 Video Retrieval System. In *Proceedings of TRECVID 2004*, Gaithersburg, MD, USA.

Arampatzis, A. and van Hameran, A. (2001). The score-distributional threshold optimization for adaptive binary classification tasks. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 285–293, New Orleans, LA, USA.

Aslam, J. A. and Montague, M. (2001). Models for metasearch. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 276–284, New Orleans, LA, USA.

Ault, T. G. and Yang, Y. (2002). Information filtering in trec-9 and tdt-3: A comparative analysis. *Information Retrieval*, 5(2-3):159–187.

Babaguchi, N., Kawai, Y., and Kitahashi, T. (2002). Event based indexing of broadcasted sports video by intermodal collaboration. *IEEE Transactions on Multimedia*, 4(1):68–75.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Addison-Wesley Harlow, England.

Bailer, W. and Schallauer, P. (2006). The Detailed Audiovisual Profile: Enabling Interoperability between MPEG-7 based Systems. In *12th International Conference on MultiMedia Modelling (MMM 2006)*, pages 217–224, Bejing, China.

Bartell, B., Cotrell, G., and Belew, R. (1994). Automatic combination of multiple ranked retrieval systems. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*, pages 173–181, Dublin, Ireland.

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D., Frieder, O., and Goharian, N. (2004). Fusion of effective retrieval strategies in the same information retrieval system. *Journal of the American Society for Information Science and Technology*, 55(10):859–868.

Belkin, N. J., Cool, C., Croft, W. B., and Callan, J. P. (1993). Effect of multiple query representations on information retrieval system performance. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '93)*, pages 339–346, Pittsburgh, PA, USA.

Belkin, N. J., Kantor, P., Fox, E. A., and Shaw, J. A. (1995). Combining the evidence of multiple query representations for information retrieval. *Information Processing and Management*, 31(3):431–448.

Bellman, R. (1961). *Adaptive control processes: a guided tour*. Princeton University Press, Princeton, N.J., USA.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is nearest neighbor meaningful. In *7th International Conference on Database Theory (ICDT 1999)*, pages 217–235, Jerusalem, Israel.

Blanken, H. M., Vries, A. P. d., Blok, H. E., and Feng, L. (2007). *Multimedia Retrieval (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Buckland, M. and Gey, F. (1994). The relationship between recall and precision. *Journal of the American Society for Information Science and Technology*, 45(1):12–19.

Byrne, D., Wilkins, P., Jones, G., Smeaton, A. F., and O'Connor., N. (2008). Measuring the impact of temporal context on video retrieval. In *Proceedings of the 7th ACM international Conference on Image and Video Retrieval (CIVR '08)*, Niagara Falls, Canada.

Cao, J., Zhang, Y.-D., Feng, B.-L., Hua, X.-F., Bao, L., Zhang, X., and Li, J.-T. (1998). TRECVID 2008 Search by MCG-ICT-CAS. In *Proceedings of TRECVID 2008*, Gaithersburg, MD, USA.

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pages 129–136, Corvallis, OR, USA.

Chang, S.-F., Sikora, T., and Puri, A. (2001). Overview of the MPEG-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695.

Chang, S.-K. and Hsu, A. (1992). Image information systems: where do we go from here? *IEEE Transactions on Knowledge and Data Engineering*, 4(5):431–442.

Chua, T.-S., Neo, S.-Y., Li, K.-Y., Wang, G., Shi, R., Zhao, M., and Xu, H. (2004). TRECVID 2004 search and feature extraction task by NUS PRIS. In *Proceedings of TRECVID 2004*, Gaithersburg, MD, USA.

Cleverdon, C. W., Mills, J., and Keen, M. (1966). Factors determining the performance of indexing systems. Technical report, ASLIB Cranfield project, Cranfield.

Clough, P., Grubinger, M., Hanbury, A., and Müller, H. (2008). Overview of the imageclef 2007 photographic retrieval task. In *Proceedings of the CLEF 2007 Workshop*, LNCS, Budapest, Hungary.

Cooke, E., Ferguson, P., Gaughan, G., Gurrin, C., Jones, G., le Borgne, H., Lee, H., Marlow, S., Donald, K. M., McHugh, M., Murphy, N., O'Connor, N., O'Hare, N., Rothwell, S., Smeaton, A. F., and Wilkins., P. (2004). TRECVID 2004 Experiments in Dublin City University. In *Proceedings of TRECVID 2004*, Gaithersburg, MD, USA.

Croft, W. B. (2000). Combining approaches to information retrieval. *Advances in Information Retrieval*, pages 1–36.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 299–306, Tampere, Finland.

Das-Gupta, P. and Katzer, J. (1983). A study of the overlap among document representations. *SIGIR Forum*, 17(4):106–114.

Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60.

Datta, R., Li, J., and Wang, J. Z. (2005). Content-based image retrieval: approaches and trends of the new age. In *Proceedings of the 7th ACM SIGMM International*

Workshop on Multimedia Information Retrieval (MIR 2005), pages 253–262, Singapore, Singapore.

de Sande, K. E. A. V., Gevers, T., and Snoek, C. G. M. (2008). Evaluation of Color Descriptors for Object and Scene Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, Anchorage, Alaska, USA.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001). Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web (WWW10)*, pages 613–622, Hong Kong, China.

Fagin, R. (1996). Combining fuzzy information from multiple systems. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1996)*, pages 216–226, Montreal, Canada.

Fletcher, R. (1987). *Practical methods of optimization; (2nd ed.)*. Wiley-Interscience, New York, NY, USA.

Fox, E. A. and Shaw, J. A. (1994). Combination of Multiple Searches. In *Proceedings of the 3rd Text REtrieval Conference (TREC-2)*, Gaithersburg, MD, USA.

Geng, X., Liu, T.-Y., Qin, T., Arnold, A., Li, H., and Shum, H.-Y. (2008). Query dependent ranking using k-nearest neighbor. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, pages 115–122, Singapore, Singapore.

Geng, X., Liu, T.-Y., Qin, T., and Li, H. (2007). Feature selection for ranking. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, pages 407–414, Amsterdam, The Netherlands.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

Harman, D. (1993). The third text retrieval conference (trec-3) january 1994 - november 1994. *SIGIR Forum*, 27(3):19–23.

Hauff, C., Murdock, V., and Baeza-Yates, R. (2008). Improved query difficulty prediction for the web. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*, pages 439–448, Napa Valley, California, USA.

Hauptmann, A. G. (2005). Lessons for the future from a decade of informedia video analysis research. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR 2005)*, pages 1–10, Singapore, Singapore.

Hauptmann, A. G. and Christel, M. G. (2004). Successful approaches in the trec video retrieval evaluations. In *Proceedings of the 12th annual ACM international conference on Multimedia (MULTIMEDIA '04)*, pages 668–675, New York, NY, USA.

Hawking, D. and Robertson, S. (2003). On collection size and retrieval effectiveness. *Information Retrieval*, 6(1):99–105.

Hoi, C.-H., Chan, C.-H., Huang, K., Lyu, M., and King, I. (2004). Biased support vector machine for relevance feedback in image retrieval. *Proceedings of the 2004 IEEE International Symposium on Neural Networks (ISSN2004)*, 4:3189–3194.

Hong, P., Tian, Q., and Huang, T. S. (2000). Incorporate support vector machines to content-based image retrieval with relevant feedback. In *International Conference on Image Processing (ICIP 2000)*, Vancouver, BC, Canada.

Huijbregts, M., Ordelman, R., and de Jong, F. (2007). Annotation of heterogeneous multimedia content using automatic speech recognition. In *Proceedings of the Second International Conference on Semantics And digital Media Technologies (SAMT 2002)*, Berlin, Germany.

Humphreys, G. W. and Bruce, V. (1989). *Visual cognition: computational, experimental and neuropsychological perspectives.* Erlbaum, Hove (U.K.).

Järvelin, A., Wilkins, P., Adamek, T., Airio, E., Jones, G., Smeaton, A. F., and Sormunen, E. (2007). DCU and UTA at ImageCLEFPhoto 2007. In *Proceedings of the CLEF 2007 Workshop*, Budapest, Hungary.

Jeong, S., Kim, K., Chun, B., Lee, J., and Bae, Y. J. (1999). An effective method for combining multiple features of image retrieval. In *TENCON 99. Proceedings of the IEEE Region 10 Conference*, volume 2, pages 982–985, Silla Cheju, Cheju Island, Korea.

Jiang, W., Zavesky, E., Chang, S.-F., and Loui, A. (2008). Cross-Domain Learning Methods for High-Level Visual Concept Classification. In *International Conference on Image Processing (ICIP 2008)*, San Diego, California, USA.

Jiang, Y.-G., Ngo, C.-W., and Yang, J. (2007). Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international Conference on Image and Video Retrieval (CIVR '07)*, pages 494–501, Amsterdam, The Netherlands.

Jin, X. and French, J. C. (2003). Improving image retrieval effectiveness via multiple queries. In *Proceedings of the 1st ACM international workshop on Multimedia databases (MMDB '03)*, pages 86–93, New Orleans, LA, USA.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02)*, pages 133–142, Edmonton, Alberta, Canada.

Kennedy, L., Chang, S., and Natsev, A. (2008). Query-Adaptive Fusion for Multimodal Search. *Proceedings of the IEEE*, 96(4):567–588.

Kennedy, L. S., Natsev, A. P., and Chang, S.-F. (2005). Automatic discovery of query-class-dependent models for multimodal search. In *Proceedings of the 13th* *annual ACM international conference on Multimedia (MULTIMEDIA '05)*, pages 882–891, Singapore, Singapore.

Kludas, J., Bruno, E., and Marchand-Maillet, S. (2008). Information Fusion in Multimedia Information Retrieval. *Adaptive Multimedial Retrieval: Retrieval, User, and Semantics: 5th International Workshop, AMR 2007, Paris, France, July 5-6, 2007 Revised Selected Papers*, pages 147–159.

Kraaij, W., Smeaton, A. F., and Over, P. (2004). TRECVID 2004 - An Introduction. In *Proceedings of TRECVID 2004*, Gaithersburg, MD, USA.

Lee, J. H. (1997). Analyses of multiple evidence combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97)*, pages 267–276, Philadelphia, Pennsylvania, USA.

Liu, T.-Y. (2008). Learning to Rank for Information Retrieval. SIGIR 2008 Tutorial.

Liu, T. Y., Xu, J., Qin, T., Xiong, W., and Li, H. (2007a). Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*.

Liu, W., Jiang, W., and Chang, S.-F. (2008a). Relevance aggregation projections for image retrieval. In *Proceedings of the 7th ACM international Conference on Image and Video Retrieval (CIVR '08)*, pages 119–126, Niagara Falls, Canada.

Liu, Y., Mei, T., Qi, G., Wu, X., and Hua, X.-S. (2008b). Query-independent learning for video search. In *IEEE International Conference on Multimedia and Expo (ICME 2008)*, pages 1249–1252, Bejing, China.

Liu, Y., Mei, T., Tang, J., Wu, X., and Hua, X.-S. (2009). Graph-based pairwise learning to rank for video search. In *15th International Conference on MultiMedia Modelling (MMM 2009)*, pages 175–184, Sophia-Antipolis, France.

Liu, Y.-T., Liu, T.-Y., Qin, T., Ma, Z.-M., and Li, H. (2007b). Supervised rank aggregation. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)*, pages 481–490, Banff, Alberta, Canada.

Luan, H., Zheng, Y., Neo, S.-Y., Zhang, Y., Lin, S., and Chua, T.-S. (2008). Adaptive multiple feedback strategies for interactive video search. In *Proceedings of the 7th ACM international Conference on Image and Video Retrieval (CIVR '08)*, pages 457–464, Niagara Falls, Canada.

Manjunath, B., Salembier, P., and Sikora, T., editors (2002). *Introduction to MPEG-7: Multimedia Content Description Language.* Wiley.

Manmatha, R., Rath, T., and Feng, F. (2001). Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 267–275, New Orleans, LA, USA.

McCabe, M., Chowdhury, A., Grossman, D., and Frieder, O. (2001). System fusion for improving performance in information retrieval systems. In *Proceedings of International Conference on Information Technology: Coding and Computing (ITCC 2001)*, Las Vegas, NV, USA.

McClave, J. T. and Sincich, T. (2006). *Statistics (10th Edition).* Prentice Hall.

McDonald, K. (2005). *Discrete Language Models for Video Retrieval.* PhD Thesis, Dublin City University, Dublin, Ireland.

McDonald, K. and Smeaton, A. F. (2005). A comparison of score, rank and probability-based fusion methods for video shot retrieval. In *Proceedings of the 4th ACM international Conference on Image and Video Retrieval (CIVR '05)*, Dublin, Ireland.

McGill, M., Koll, M., and Noreault, T. (1979). An evaluation of factors affecting document ranking by information retrieval systems. Technical Report NSF-IST-78-10454 to the National Science Foundation (USA), Syracuse University.

Mei, T., Zheng-Jun, Z., Liu, Y., Wang, M., Qi, G.-J., Tian, X., Wang, J., Yang, L., and Hua, X.-S. (2008). MSRA at TRECVID 2008. In *Proceedings of TRECVID 2008*, Gaithersburg, MD, USA.

Metzler, D. and Croft, W. B. (2007). Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274.

Metzler, D., Strohman, T., and Croft, W. B. (2006). Indri at TREC 2006: Lessons Learned From Three Terabyte Tracks. In *Proceedings of the 15th Text REtrieval Conference (TREC-14)*, Gaithersburg, MD, USA. electronic proceedings only.

Montague, M. and Aslam, J. A. (2001). Relevance score normalization for metasearch. In *Proceedings of the 10th international conference on Information and knowledge management (CIKM '01)*, pages 427–433, Atlanta, Georgia, USA.

Montague, M. and Aslam, J. A. (2002). Condorcet fusion for improved retrieval. In *Proceedings of the 11th international conference on Information and knowledge management (CIKM '02)*, pages 538–548, McLean, Virginia, USA.

MPEG-7 (2001). Multimedia Content Description Interface. Standard No. ISO/IEC n?15938.

Naphade, M., Smith, J. R., Tesic, J., Chang, S.-F., Hsu, W., Kennedy, L., Hauptmann, A., and Curtis, J. (2006). Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86–91.

Natsev, A. P., Naphade, M. R., and Tesic, J. (2005). Learning the semantics of multimedia queries and concepts from a small number of examples. In *Proceedings of the 13th annual ACM international conference on Multimedia (MULTIMEDIA '05)*, pages 598–607, Singapore, Singapore.

O'Connor, N., Cooke, E., le Borgne, H., Blighe, M., and Adamek., T. (2005). The AceToolbox: Low-Level Audiovisual Feature Extraction for Retrieval and Classification. In *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*, London, U.K.

Ogilvie, P. and Callan, J. (2003). Combining document representations for known-item search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, pages 143–150, Toronto, Canada.

Ounis, I., Lioma, C., Macdonald, C., and Plachouras, V. (2007). Research directions in terrier. *Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper.*

Over, P., Awad, G., Kraaij, W., and Smeaton, A. F. (2007). TRECVID 2007 Overview. In *Proceedings of TRECVID 2007*, Gaithersburg, MD, USA.

Over, P., Awad, G., Rose, T., Fiscus, J., Kraaij, W., and Smeaton, A. F. (2008). TRECVID 2008 - Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *Proceedings of TRECVID 2008*, Gaithersburg, MD, USA.

Renda, M. E. and Straccia, U. (2003). Web metasearch: rank vs. score based rank aggregation methods. In *Proceedings of the 2003 ACM symposium on Applied computing (SAC '03)*, pages 841–846, Melbourne, FL, USA.

Robertson, S. (2007). On score distributions and relevance. In *29th European Conference on Information Retrieval (ECIR 2007)*, pages 40–51, Rome, Italy.

Robertson, S., Zaragoza, H., and Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. In *Proceedings of the 13th international conference on Information and knowledge management (CIKM '04)*, pages 42–49, Washington, DC, USA.

Robertson, S. E. (1977). The probability ranking principle in information retrieval. *Journal of Documentation*, (33):294–304.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall Series in Automatic Computation, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs NJ.

Rui, Y., Huang, T., and Mehrotra, S. (1997). Content-based image retrieval with relevance feedback in MARS. *Image Processing, 1997. Proceedings., International Conference on*, 2:815–818 vol.2.

Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62.

Ruthven, I. and Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145.

Sadlier, D., Marlow, S., O'Connor., N., and Murphy., N. (2002). Automatic tv advertisement detection from mpeg bitstream. *Journal of the Pattern Recognition Society, Vol.35, No.12 (ISSN: 0031-3203)*, 35(12):2719–2726.

Sadlier, D., O'Connor, N., Marlow, S., , and Murphy., N. (2003). A combined audio-visual contribution to event detection in field sports broadcast video. case study: Gaelic football. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '03)*, pages 552–555.

Salton, G. (1989). *Automatic Text Processing*. Addison–Wesley.

Santini, S. and Dumitrescu, A. (2008). Context and activity games as a non-ontological model of semantics. In *Proceedings of the Third International Conference on Semantics And digital Media Technologies (SAMT 2003)*, Koblenz, Germany.

Saracevic, T. (2007). Relevance: A review of the literature and a framework for thinking on the notion in information science. Part III: Behavior and effects of relevance. *Journal of the American Society for Information Science and Technology (JASIST)*, 58(13):2126–2144.

Saracevic, T. and Kantor, P. (1988). A study of information seeking and retrieving, iii: Searchers, searches, overlap. *Journal of the American Society for Information Science and Technology (JASIST)*, 39:177–196.

Savoy, J., Calvé, A. L., and Vrajitoru, D. (1996). Report on the trec-5 experiment: Data fusion and collection fusion. In *Proceedings of the 5th Text REtrieval Conference (TREC-4)*, Gaithersburg, MD, USA.

Smeaton, A. F. (1998). Independence of contributing retrieval strategies in data fusion for effective information retrieval. In *BCS-IRSG Annual Colloquium on IR Research*, Autrans, France.

Smeaton, A. F. (2004). *ARIST - Annual Review of Information Science and Technology*, volume 38, chapter 8. Indexing, Browsing and Searching of Digital Video, pages 371–407. American Society for Information Science and Technology.

Smeaton, A. F., Kraaij, W., and Over, P. (2003). TRECVID 2003 - An Overview. In *Proceedings of TRECVID 2003*, Gaithersburg, MD, USA.

Smeaton, A. F., Over, P., and Doherty, A. (2009). Video Shot Boundary Detection: Seven Years of TRECVid Activity. *Computer Vision and Image Understanding.*

Smeaton, A. F., Over, P., and Kraaij, W. (2006). Evaluation Campaigns and TRECVid. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia information retrieval (MIR 2006).*

Smeaton, A. F. and Quigley, I. (1996). Experiments on using semantic distances between words in image caption retrieval. In *Proceedings of the 19th Annual Inter-*

national ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96), pages 174–180, Zurich, Switzerland.

Smeaton, A. F., Wilkins, P., Worring, M., de Rooij, O., Chua, T.-S., and Luan, H. (2008). Content-based video retrieval: Three example systems from TRECVid. *International Journal of Imaging Systems Technology*, 18(2-3):195–201.

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380.

Snoek, C., van de Sande, K., de Rooij, O., Huurnink, B., van Gemert, J., Uijlings, J., He, J., Li, X., Everts, I., Nedovic, V., van Liempt, M., van Balen, R., Yan, F., Tahir, M., Mikolajczyk, K., Kittler, J., de Rijke, M., Geusebroek, J., Gevers, T., Worring, M., Smeulders, A., and Koelma, D. (2008). The MediaMill TRECVID 2008 Semantic Video Search Engine. In *Proceedings of TRECVID 2008*, Gaithersburg, MD, USA.

Snoek, C. G. M., Worring, M., and Smeulders, A. W. M. (2005). Early versus Late Fusion in Semantic Video Analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia (MULTIMEDIA '05)*, pages 399–402, Singapore, Singapore.

Snoek, C. G. M., Worring, M., van Gemert, J. C., Geusebroek, J.-M., and Smeulders, A. W. M. (2006). The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia. In *Proceedings of the 14th annual ACM international conference on Multimedia (MULTIMEDIA '06)*, pages 421–430, Santa Barbara, CA, USA.

Swets, J. A. (1963). Information retrieval systems. *Science*, 141:245–250.

Swets, J. A. (1969). Effectiveness of information retrieval methods. *American Documentation*, (20):72–89.

Tešić, J., Natsev, A., Xie, L., and Smith, J. (2007a). Data modeling strategies for imbalanced learning in visual search. In *IEEE International Conference on Multimedia and Expo (ICME 2007)*, pages 1990–1993, Hannover, Germany.

Tešić, J., Natsev, A. P., and Smith, J. R. (2007b). Cluster-based data modeling for semantic video search. In *Proceedings of the 6th ACM international Conference on Image and Video Retrieval (CIVR '07)*, pages 595–602, Amsterdam, The Netherlands.

Tong, S. and Chang, E. (2001). Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia (MULTIMEDIA '01)*, pages 107–118, Ottawa, Canada.

Turtle, H. and Croft, W. (1991). Evaluation of an Inference Network-based Retrieval Model. *ACM Transactions on Informaion Systems*, 9(3):187–222.

Urban, J. and Jose, J. M. (2004). Evidence Combination for Multi-Point Query Learning in Content-Based Image Retrieval. In *Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISME 2004)*, pages 583–586, Washington, DC, USA.

Van Rijsbergen, C. (1979). *Information Retrieval, 2nd edition.* Butterworth-Heinemann Newton, MA, USA.

Vapnik, V. N. (1995). *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA.

Vogt, C. C. and Cottrell, G. W. (1999). Fusion Via a Linear Combination of Scores. *Information Retrieval*, 1(3):151–173.

Voorhees, E. M., Gupta, N. K., and Johnson-Laird, B. (1995). Learning collection fusion strategies. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '95)*, pages 172–179, Seattle, Washington, USA.

Westerveld, T. (2004). *Using generative probabilistic models for multimedia retrieval.* Phd thesis, University of Twente, Enschede, The Netherlands.

Whitten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques; (2nd ed.).* Morgan Kaufmann Publishers, San Francisco, CA, USA.

Wilkins, P. (2007). Automatic query-time generation of retrieval expert coefficients for multimedia retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, pages 924–924, Amsterdam, The Netherlands.

Wilkins, P., Adamek, T., Jones, G., O'Connor, N., and Smeaton., A. F. (2007a). Trecvid 2007 experiments at dublin city university. In *Proceedings of TRECVID 2007*, Gaithersburg, MD, USA.

Wilkins, P., Adamek, T., O'Connor, N., and Smeaton, A. F. (2007b). Inexpensive Fusion Methods for Enhancing Feature Detection. *Signal Processing: Image Communication, Special Issue on Content-Based Multimedia Indexing and Retrieval*, 22(7-8):635–650.

Wilkins, P. and et al. (2007). KSpace at TRECVid 2007. In *Proceedings of TRECVID 2007*, Gaithersburg, MD, USA.

Wilkins, P., Ferguson, P., Gurrin, C., and Smeaton., A. F. (2006a). *Automatic Determination of Feature Weights for Multi-Feature CBIR*, volume 3936 / 2006, pages 527–530. Springer, Berlin / Heidelberg, Germany.

Wilkins, P., Ferguson, P., and Smeaton, A. F. (2006b). Using Score Distributions for Query-time Fusion in Multimedia Retrieval. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia information retrieval (MIR 2006)*, pages 51–60.

Wilkins, P., Troncy, R., Halvey, M., Byrne, D., Amin, A., Punitha, P., Smeaton, A. F., and Villa, R. (2009). User variance and its impact on video retrieval benchmarking. In *Proceedings of the 8th ACM international Conference on Image and Video Retrieval (CIVR '09)*, Santorini, Greece.

Witten, I., Moffat, A., and Bell, T. C. (1999). *Managing gigabytes: compressing and indexing documents and images.* Morgan Kaufmann Publishers Inc.

Xia, F., Liu, T.-Y., Wang, J., Zhang, W., and Li, H. (2008). Listwise approach to learning to rank: theory and algorithm. In *Helsinki, Finland*, pages 1192–1199, Helsinki, Finland.

Xie, L., Natsev, A., and Tesic, J. (2007). Dynamic Multimodal Fusion in Video Search. In *IEEE International Conference on Multimedia and Expo (ICME 2007)*, pages 1499–1502, Hannover, Germany.

Yan, R. and Hauptmann, A. G. (2003). The combination limit in multimedia retrieval. In *Proceedings of the eleventh ACM international conference on Multimedia (MULTIMEDIA '03)*, pages 339–342, Berkeley, CA, USA.

Yan, R. and Hauptmann, A. G. (2006). Probabilistic latent query analysis for combining multiple retrieval sources. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 324–331, Seattle, Washington, USA.

Yan, R., Hauptmann, A. G., and Jin, R. (2003). Negative pseudo-relevance feedback in content-based video retrieval. In *Proceedings of the eleventh ACM international conference on Multimedia (MULTIMEDIA '03)*, pages 343–346, Berkeley, CA, USA.

Yan, R., Yang, J., and Hauptmann, A. G. (2004). Learning query-class dependent weights in automatic video retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia (MULTIMEDIA '04)*, pages 548–555, New York, NY, USA.

Yanagawa, A., Chang, S.-F., Kennedy, L., and Hsu, W. (2007). Columbia University's Baseline Detectors for 374 LSCOM Semantic Visual Concepts. Technical report, Columbia University, New York, NY, USA.

Zhang, H., Tan, S. Y., Smoliar, S. W., and Yihong, G. (1995). Automatic parsing and indexing of news video. *Multimedia Systems*, 2(6):256–266.

Zhang, Y. and Callan, J. (2001). Maximum likelihood estimation for filtering thresholds. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 294–302, New Orleans, LA, USA.

Zhou, X. S. and Huang, T. S. (2003). Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544.

Zhou, Y. and Croft, W. B. (2006). Ranking robustness: a novel framework to predict query performance. In *Proceedings of the 15th international conference on Information and knowledge management (CIKM '06)*, pages 567–574, Arlington, VA, USA.

Zhou, Y. and Croft, W. B. (2007). Query performance prediction in web search environments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, pages 543–550, Amsterdam, The Netherlands.