AUTOSEEK

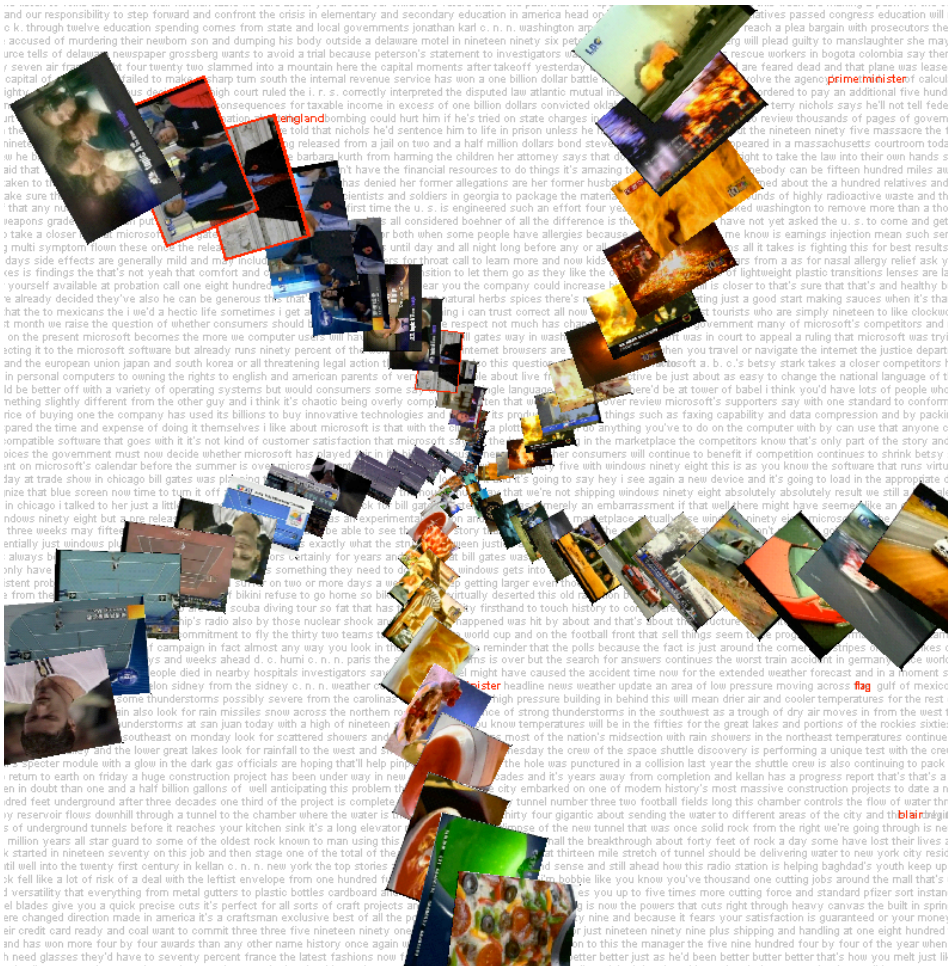
TOWARDS A FULLY AUTOMATED VIDEO SEARCH SYSTEM

A Thesis Presented by

Bouke Huurnink

to

de Faculteit der Natuurwetenschappen, Wiskunde en Informatica
in Partial Fulfilment of the Requirements for the Degree of Master of Science
in the Subject of Information Sciences (Multimedia)



Universiteit van Amsterdam
Amsterdam, the Netherlands
October 2005

Advisor: Dr. Marcel Worring

Abstract

The astounding rate at which digital video is becoming available has stimulated research into video retrieval systems that incorporate visual, auditory, and spatio-temporal analysis. In the beginning, these multimodal systems required intensive user interaction, but during the past few years automatic search systems that need no interaction at all have emerged, requiring only a string of natural language text and a number of multimodal examples as input. We apply ourselves to this task of automatic search, and investigate the feasibility of automatic search without multimodal examples. The result is AutoSeek, an automatic multimodal search system that requires only text as input. In our search strategy we first extract semantic concepts from text and match them to semantic concept indices using a large lexical database. Queries are then created for the semantic concept indices as well as for indices that incorporate ASR text. Finally, the result sets from the different indices are fused with a combination strategy that was created using a set of development search statements. We subject our system to an external assessment in the form of the TRECVID 2005 automatic search task, and find that our system performs competitively when compared to systems that also use multimodal examples, ranking in the top three systems for 25% of the search tasks and scoring the fourth highest in overall mean average precision. We conclude that automatic search without using multimodal examples is a realistic task, and predict that performance will improve further as semantic concept detectors increase in quantity and quality.

Table of Contents

Acknowledgements

In the summer of ’98 I leaped from a life in a small New Zealand country town into life in the big city as a student of engineering at the University of Auckland. It quickly became apparent that city life agreed with me. Engineering on the other hand did not, so I put university on hold and went exploring. A few years and a few countries later I was ready to go back to formal education, so I started to look for a program that would allow me to cover a broad range of subjects. It was at this point that I discovered the cross-disciplinary Information Sciences degree at the University of Amsterdam, a degree aimed at investigating the psychological, economic, and technological implications of the ever-changing world of technology that surrounds us. It sounded like me, so I went ahead and made another leap.

That was 2001. This is 2005, and I am glad to say that I made the right choice. Before you lies my thesis, which represents the culmination of the years that I have spent studying here in Amsterdam. I was lucky to be able to join forces with the MediaMill research group and participate in their efforts towards machine understanding of video.

I would like to take this opportunity to thank the people that helped me along the way to producing this thesis:

- Marcel Worring, who as my thesis supervisor was a source of boundless enthusiasm, and managed to pick me up and dust me off whenever I stumbled over the more esoteric elements involved in my research.
- Cees Snoek, the “semantic chef”, who developed the concept detectors I use, always had tips for papers to read, had new ideas for my strategy, and one day made a deal with me: if I participated in the TRECVID common annotation effort, I could implement my research and enter one of MediaMill’s seven runs. I had no idea what I was in for.
- The MediaMill team members, who were always supportive and ready to share their knowledge. I would especially like to thank Jan van Gemert for his role in creating the semantic concept detectors and also for creating an implementation of his LSI tool that was compatible with my system, and Dennis Koelma for creating a visualiser for results that to me had just been a list of numbers on a screen.
- Maarten de Rijke, who created a challenging course on Information Retrieval that may have slowed me down, but which proved to be of great help in forming my ideas about multimodal retrieval.
- Kwin and Merel, who were excellent sounding boards, offered sage advice, and made great dinners.
- My friends and family, for supporting me throughout all of this.
- And of course Adam, who dedicated many of the few hours of his free time to reading this thesis and helped me out in more ways than one.

Now, on to the next leap…

CHAPTER 1 INTRODUCTION

Video has reached critical mass in the digital revolution. Widespread availability of affordable digital video technology, accompanied by exponential increases in digital storage capacity and semi-conductor processing power, means that home users can now potentially create, store, and access unlimited hours of video on home systems. A short inventory of digital video products currently available on the consumer market includes digital video cameras, hard disks dedicated to capturing television data, and digital video processing software. There is even the recently emerged "video walkman", which allows people to view digital video on the train, in the car, on the beach, anywhere. Increasing bandwidth availability also means that more and more digital video is being disseminated on the Internet. It is no longer unusual for commercial television channels to make large portions of their broadcasting available online and for free.

The result? An explosion in the availability of digital video material; an explosion that shows no signs of slowing. And that proliferation of video data brings with it the burden seemingly peculiar to post-Internet society: the problem of *too much information*. With enormous quantities of digital video information at our disposal, how are we to locate that one home video of the lions when we went to Africa years ago? Or a video clip showing fireworks for the montage we are making? And what about that news item featuring a well-known political leader misspeaking at a recent press conference? With an abundance of video available to us, we have to be able to search quickly and efficiently to retrieve the item we need.

The problem of information retrieval is not a new one. The introduction of the Internet has led to extensive research efforts into the retrieval of digital information. We see the fruits of these labours in the many commercial applications that are now available, allowing us to search hard drives, the Internet, our library catalogues, and more. However, these research efforts have been directed primarily at searching through text. With the introduction of video content into the digital arena, we are faced with a problem that our advances in text retrieval have not fully prepared us to face. How do we search through video, a medium that that does not include a textual modality?

International research efforts are equipping us with a number of tools that help us solve this problem. Automated Speech Recognition (**ASR**)[1] technology allows us to transcribe dialogue and search it as text. Advances in **low-level feature** detection allow us to "query by example". This means that we can provide a system with an example video clip, and the system will then search for video clips that have similar characteristics, including visual characteristics (e.g. colour), auditory characteristics (e.g. sound energy patterns), and spatio-temporal aspects (e.g. motion). We can even detect a limited number of semantic concepts as they occur in video – specific objects, people, and settings that have some kind of integral meaning to human beings. Think, for example, of concepts such as "explosion", "Bill Clinton", or "forest", which have no inherent meaning to machines but are instantly recognisable to most human beings.

The video search engines that have resulted from these efforts are still in their infancy. They often offer a large variety of options, allowing users to search on video aspects as diverse as motion features, textual content and (limited) semantic content. The advantage of the diversity of options is that the user has a large amount of control in deciding what kind of questions to ask the system, and which aspects of the video collection they wish to explore. The disadvantage of interactive searches is that they can be time-consuming and complicated. It is not realistic to expect users to spend a large amount of time iterating through different queries to try to find relevant video clips. We need a system that creates queries automatically, one that returns optimal results without requiring a lot of work.

1.1 Multimodal Video Search

Our research takes place within the domain of the **multimodal video search engine**, which we define as follows:

> **Definition 1. (Multimodal Video Search Engine)** *A system that allows users to retrieve videos from a database according to some criteria, and allows them to incorporate more than one mode (visual, auditory, and spatio-temporal) of video in that search.*

A typical multimodal search engine will allow users to enter text to search for in ASR transcripts, enter examples of the type of video that they are looking for, request videos with certain colour, motion, or sound characteristics, and to search for semantic concepts such as "car" or "building" using specialised detectors that

[1] Throughout this paper, terms defined in Appendix I are printed in bold face at initial usage.

analyse many different aspects of video. Multimodal video search engines should not be confused with engines that operate using only one mode of video, such as the Blinkx.TV video search engine[2], which searches only on auditory information in the form of ASR speech transcripts. Multimodal video search engines should also be distinguished from video search engines that use information not directly contained in video, such as the Yahoo! video search engine[3], which utilises information from text surrounding video links on web pages, or the Google video search engine[4], which combines information from surrounding text, captions and when they are available, manually created transcripts.

Multi-modal video search engines generally require intensive interaction, placing a high level of cognitive load on the user in the search process. When a user is faced with many search options, it is a matter of trial and error to find the combination of queries that deliver the desired results. In this thesis, a search that require this type of user involvement will be termed an **interactive search**, illustrated in Figure 1. In the interactive search the user first analyses what kind of information is required, decides which kind of queries might be appropriate, and performs an initial search. The user then analyses the search results and refines the queries as necessary. This process is repeated until the desired results are found.

> **Definition 2. (Interactive Search)** *A search in which the user iteratively interacts with a search system over an extended period of time, typically composing queries in different modalities and interactively helping the search system to rank results.*

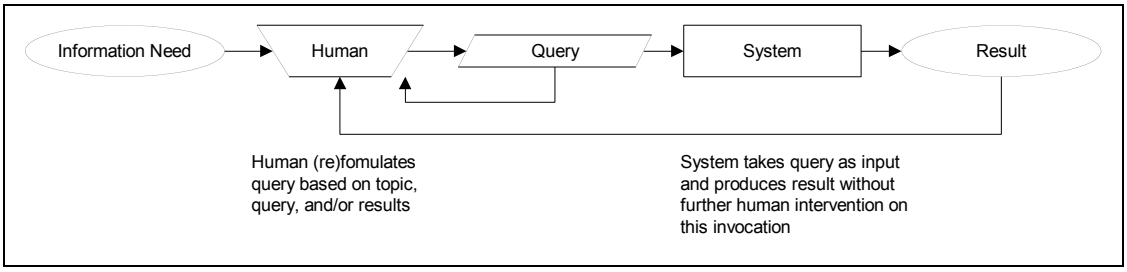


Figure 1. The Interactive Search process[5]

For multimodal video search technology to become commonly accepted, it is necessary for it to be as easy to use as possible. We aim to minimise the load on users by eliminating their involvement in the search process. Shown in Figure 2, the **automatic search** places the burden of query development and result selection on the system, rather than the user. The system is presented with some sort of information need, develops the appropriate queries, and returns a result set answering that information need.

> **Definition 3. (Automatic Search)** *A search in which the user enters only an initial information need: subsequent query selection, expansion and result ranking is done by the search system without any user intervention.*

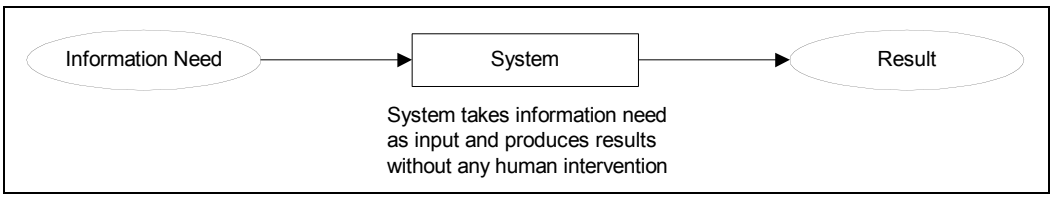


Figure 2. The Automatic Search process

[2] http://www.blinkx.tv/
[3] http://video.yahoo.com/
[4] http://video.google.com/
[5] Diagrams of interactive and automatic search processes adapted from TRECVID 2005 guidelines at http://www-nlpir.nist.gov/projects/tv2005/tv2005.html.

An example of such a search is shown in Figure 3. In this example, the information that is needed is shots of Bill Clinton standing in front of the American flag. This is communicated to the system by means of a textual statement and a number of example video shots. The multimodal video search system then develops a number of queries tailored to the information that it contains. In this example, the system contains specialised detectors for the semantic concepts "Bill Clinton" and "flag", as well as a module that calculates the similarity of videos in the collection to the example videos. It also contains a search engine for the speech information that has been extracted from the auditory mode through ASR.

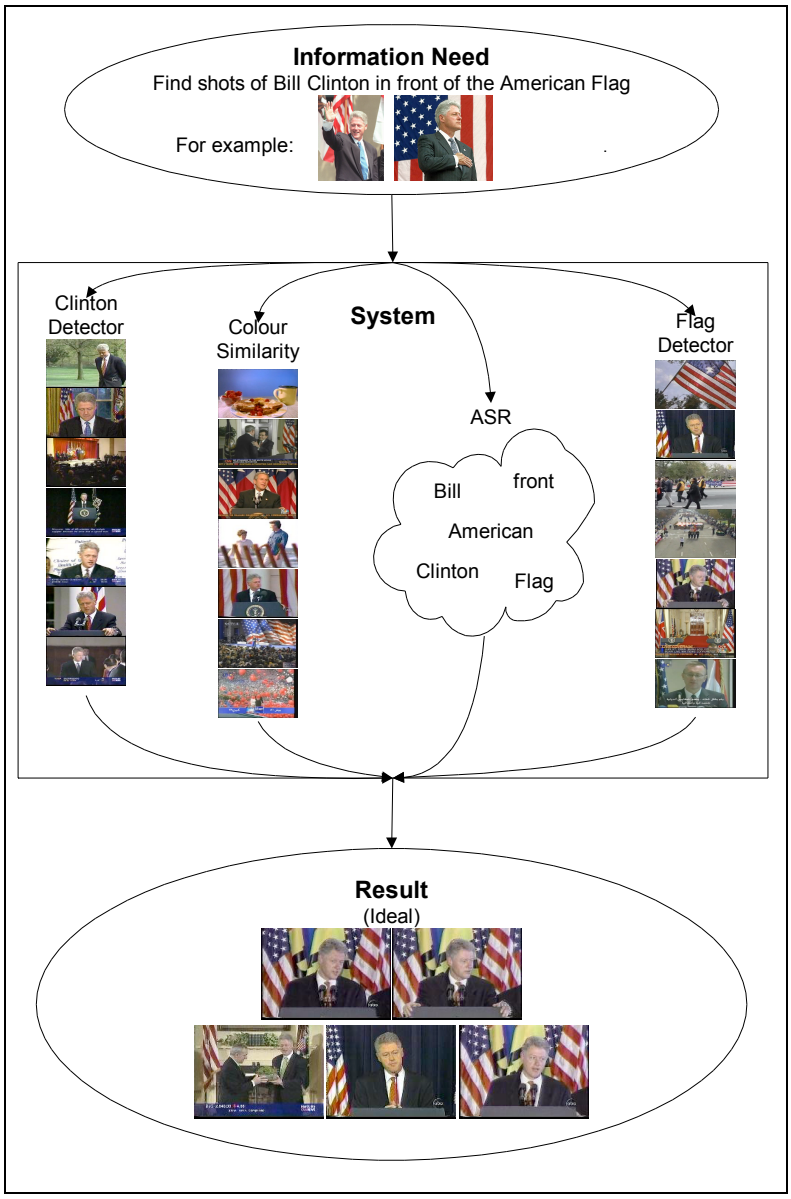


Figure 3. Example of an automatic search

1.2 Problem Statement

Automatic search has begun to attract research interest. The majority of automatic search systems to date require two types of information from the user: a number of examples of the type of video clip that is required; and an **information request**, which we define as:

> **Definition 4. (Information Request)** A natural language text statement that defines a need for some type of multimodal content.

We consider it unreasonable to require users to provide examples of the type of video clip they are searching for before they start a search. In many cases, if users already have examples of the video clip that they want, it will not be necessary to find more. It also requires effort on the part of the user to search for examples of video clips in preparation for the automatic search; and therefore an initial search is still required prior to the automatic search. This inconsistency leads us to ask the fundamental question that will be addressed in this thesis:

> ***How can we perform automatic multimodal video search using only text as input?***

To tackle this question, we will design and implement a system capable of automatic search, AutoSeek. AutoSeek will incorporate a large, fluctuating number of state-of-the-art multimodal concept detectors, as well as text search capabilities. To achieve this, we will have to translate a text request into queries for the various semantic concept detectors. This brings us to our first follow-up question:

> *How can we identify semantic concept detectors that are related to text?*

A simple word matching technique can be used to identify semantic concepts that are related to text, but are likely to be insufficient. By matching words, a system might be able to identify the concept "George Bush" as being related to the text "I would like to see clips of George Bush", but what if the request is for "shots of the president of the United States"? To successfully achieve this translation it will be necessary to find ways for the system to interpret digital data in a similar way to humans and thereby overcome the **semantic gap**, the divide between human interpretation of digital data and machine perception of the same data (Smeulders and Worring 2000). MediaMill has already created concept detectors to form an initial bridge across this divide, an d now we must bridge the gap between human interpretation of digital text and machine interpretation of the same text.

We will take a knowledge-based approach to this problem, identifying semantic concepts in request text and relating them to the available semantic concept detectors through the use of a natural-language resource. We also incorporate traditional text retrieval techniques using the spoken language transcript of the video, and use a large set of development data to optimise the different kinds of search. The final result will be a number of different sets of results. This brings us to our next follow-up question:

> *How can we combine results from heterogeneous types of searches into a single set of results?*

We will develop a strategy for combination of different kinds of search results, once again using a large set of development data to optimise and evaluate the strategy. The final result will a multimodal video search engine that is capable of automatic search using only text as input. We wish to know how AutoSeek compares to other automatic search systems, which brings us to our final question:

> *Can an automatic multimodal video search engine that uses only text as input compete with engines that use multimodal examples as well as text?*

We will submit AutoSeek to external assessment within the context of the TREC Video Retrieval Evaluation (**TRECVID**). Here the performance of AutoSeek for 24 different statements of information need is evaluated. Ten automatic multimodal video search engines that incorporate multimodal examples[6] as well as text input are evaluated using the same data set and the same statements of information need. This allows us to make an objective comparison between our system and systems that include examples, and discuss the feasibility of automatic multimodal search video search using only text as input.

[6] To be confirmed in TRECVID publications at the end of 2005.

1.3 Paper Organisation

We investigate the current state-of-the-art in automatic search in Chapter 2. We start this chapter by outlining a generic architecture that can be used to help define automatic multimodal search systems. Subsequently we investigate the strategies that others have used to realise automatic multimodal video search systems. Chapter 3 outlines our design and implementation of the AutoSeek system. We pay special attention to our semantic, knowledge-based approach to automatic search, and also outline the implementation of the AutoSeek system. It continues with a description of the overall structure of the system. In 0 we describe the development of an optimised search strategy for AutoSeek, and discuss the subsequent evaluation by an external examination in Chapter 5. We conclude the paper in Chapter 6 by addressing our initial question, *how do we perform automatic multimodal search using only text as input?*

CHAPTER 2 AUTOMATIC SEARCH: A REVIEW

Progress in multimodal search for video archives is advancing quickly, most notably through the TRECVID benchmark evaluations. In this chapter we review research from a number of different research teams. Literature covered in this chapter includes material from the LowLands team (Westerveld, Ianeva et al. 2003; Ianeva, Boldareva et al. 2004), the Informedia team from Carnegie Mellon University (Hauptmann, Chen et al. 2004; Yan, Hauptmann et al. 2004), the IBM team at TJ Watson research centre (Amir, Berg et al. 2003; Amir, Argillander et al. 2004), the Dublin City University (DCU) team (Blott, Boydell et al. 2004; Cooke, Ferguson et al. 2004), and the NUS PRIS team at the National University of Singapore (Chua, Neo et al. 2004).

2.1 A Generic Architecture for Automatic Multimodal Search

Before we analyse the specific details of other research efforts within automatic search, we define a general framework to which most automatic multimodal search engines conform. When reviewing the literature in automatic multimodal video search, a general architecture emerges, illustrated in Figure 4 below.

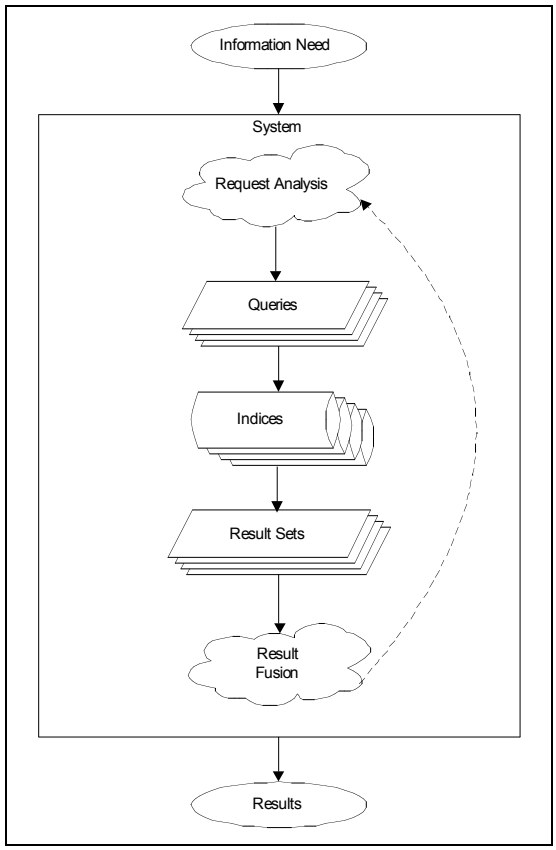


Figure 4. A generic architecture for automatic multimodal search

First, an automatic search system receives some kind of statement of information need as input. This may include a textual question, example videos, and example images. Various techniques are then used to analyse the statement. The results of the analysis are used to form a number of **queries** for the different system search indices, which may allow the system to search for certain words in speech, semantic concept features (for example, "explosion"), or for fragments that are similar to an example video clip. Each query is sent to an **index** and produces a result set containing a number of **shots**, short segments of video. The shots are usually ranked according to the likelihood that they contain the required information. Sometimes the results of a query are used to provide feedback for the system (illustrated in the diagram with a dotted line), allowing it to adjust the original analysis of the information need and reform queries. Finally, results from different queries to various indexes are combined using a fusion strategy to produce a list of integrated results. We will discuss different approaches to each of the steps in this generic model in the following sections.

2.2 Indices

Essential to any search of large amounts of data is the index, mapping query terms to relevant shots. For example, a text index usually maps words to shots that contain those words. When a query is passed to an index, the index returns shots that are said to be relevant to the query. Usually these shots are returned with a relevance value that indicates the probability that the resulting shots are exactly relevant to the query. Indices are created through analysis of the retrieval database. They are usually generated before the search process, as they can take a large amount of time to generate.

2.2.1 Low-Level Feature Indices

Low-level features describe characteristics of video that can be directly extracted from the raw digital data. They can be classified according to their modality: visual, auditory, or spatio-temporal. Low-level feature indices within automatic search are primarily used when searching by example. They allow retrieval of shots with features similar to those of an example video fragment; for instance shots with a similar colour distribution.

Visual Indices

A number of visual aspects are valuable in video search. One of the most important is the distribution of colour within a video, which can be described by statistical distributions such as colour histograms and colour correlograms. Informedia, IBM, LowLands, and Dublin City University all index the colour characteristics of shots in the collection. Other important low-level visual features include edges (DCU, Informedia, and IBM) and texture (IBM, DCU, LowLands, and Informedia).

Auditory Indices

Auditory characteristics of video are less widely utilised than visual characteristics. This is possibly due to the fact that searches through video databases are usually focussed on visual objects, rather than sounds, that occur in the video. LowLands indexes characteristics of acoustic energy at discrete time intervals, and NUS PRIS also indexes sound energy characteristics, later combining them with other features to identify speakers in the shots.

Spatio-temporal Analysis

Spatio-temporal characteristics (characteristics of video across space and time) are, like auditory characteristics, less widely used than visual characteristics in automatic video search. An important use of low-level spatio-temporal characteristics is to allow for object detection. To this end, LowLands indexes motion features. These are later used to differentiate between static and moving objects. In general, an area in video that shows cohesive movement over a period of time can be said to be an object.

2.2.2 Text Indices

Video does not contain machine readable text, yet text is a very important component of automatic video search. ASR allows us to translate the spoken words of people in video into machine-readable text. While the speech of people in video does not always describe the visual contents of a video shot it can be a very valuable clue, especially in news video data. News video often contains voice-overs and announcements describing the core subjects of the news items that are being presented. A movie, on the other hand, may consist mostly of dialogue that describes the thoughts and emotions of the characters, but not the visual contents of the video. Therefore ASR is important when searching through news video, though it may be not be quite as important when searching through other video genres. On-screen caption information from Video Optical Character Recognition (Video OCR) is also used by some systems to augment ASR.

Basic Text Indexing

Essential to all of the reviewed systems is a text index that allows the system to map words from a query to words from the ASR of a shot. When such an index is created, **stop words** are identified and removed. Stop words are words that occur so frequently in the English language that they are of little use in searches. Some obvious examples in English are words like “the”, “and”, “a”, and “of”. Sometimes the words occurring in ASR are also **stemmed**. When words are stemmed, the morphological variants of words are reduced to a common root. The words “run” and “running”, for example, will usually be reduced to the common root form “run”[7]. The Porter stemmer (Porter 1980) is a good example of a stemming algorithm, and one of the most widespread stemmers for English. This algorithm was used by Informedia and DCU. The LowLands team also employed stemming, but did not report which algorithm they used.

[7] This can vary with different stemming algorithms.

When a query consisting of a number of words, or terms, is passed to a text index, shots containing those words are retrieved from the index. The shots are then ranked according to the **inverse document frequency** and the **term frequency** of the query terms that they contain. Inverse document frequency is a measure of how often a term occurs in a collection of many documents (in the current domain, the text that belongs to a shot can be considered a document). Term frequency is a measure of how often a term occurs in a shot that is being retrieved. The text retrieval community has found these two measures to be extremely important for retrieving relevant text documents. The two measures can be explained intuitively: if a query term is very common in a collection, then it will return many documents, and therefore it has a low discriminatory value. If a query term occurs often in a shot, then the shot is more likely to be relevant. There are many ways in which inverse document frequency and term frequency can be combined (Salton and Buckley 1988). NUS PRIS, Informedia and DCU all employ the OKAPI BM25 (Robertson 1995) retrieval combination schema. IBM also uses an OKAPI–based approach. OKAPI is a system designed for searching text documents, developed at City University London. Its retrieval techniques all incorporate term frequency and inverse document frequency. It also offers a number of methods that can be used when performing automatic search of text (Robertson, Walker et al. 2000). LowLands have created their own retrieval combination scheme based on language modelling, outlined in Hiemstra (2001). Their approach also incorporates term frequency and inverse document frequency. In this thesis we will refer to indices that use term frequency and inverse document frequency as TFIDF indices.

Video Specific Augmentation of Text Indexes

The text retrieval techniques discussed above were originally developed for written documents. In a collection of written documents, each document can be viewed independently of the others. This is not the case for a video collection that has been separated into shots. Shots on their own do not form an independent story; they contain information, but have generally been edited so that they form a logical story unit only when they are viewed in sequence (Snoek and Worring 2005). It is often the case that a newscaster introduces a subject from the studio and the subject is shown later in the story, in a different shot. This results in a temporal mismatch between the text and the video. A traditional search on the ASR transcript would then return the shot where the subject is announced in the studio, and not the adjacent shots containing the subject. LowLands, Informedia, and DCU compensate for this by treating shots adjacent to the shot with the word as implicitly being part of the same story unit, with the relevance decreasing as the surrounding shots become further separated from the shot containing the query word. IBM took a different approach, incorporating specialised "story boundary" detectors that explicitly identify logical story units. These boundaries are used to help create an auxiliary index that maps the ASR data to story units. Another approach taken by IBM is to create documents of consecutive words with a length of 100, with mappings from words to shots.

2.2.3 Semantic Concept Indices

DCU and LowLands only utilise low-level feature indices and text indices in their automatic systems. IBM, Informedia, and NUS PRIS also incorporate high-level semantic concept indices. We distinguish two types of semantic concept indices. The first type of index is the detector-based index. The second type is the language-based index. Detector-based semantic concept indices are generated through the use of low-level features and the words associated with shots, while language-based indices are generated through analysis of the relationships between words in the ASR transcripts of the shots.

Detector-Based Semantic Concept Indices

A query to a detector-based index will return a list of shots, ranked by the probability that they contain a certain concept. For example, a query to the detector-based semantic index for the concept "car" will return a list of all of the shots in the video collection, ranked according to the pre-calculated probability that they contain a car. The creation of a detector requires a number of examples that can be used to help identify the characteristics of the types of video that conform to concepts. These characteristics may be in the form of low-level features, ASR text, and other semantic concepts. The detector is trained on a large data set to refine the parameters. Semantic concept detectors are not perfect, and some work better than others. IBM incorporates 46 different semantic concept indices of frequently occurring concepts. Informedia uses 10 semantic concept indices in their systems (including, for example, "people", "basketball", and "fire"). NUS PRIS uses 6 semantic concepts that correspond to 6 different types of requests that they classify (see section 2.3.2). Informedia and NUS PRIS also utilise face detectors, and place the characteristics of faces found in a separate index.

Language-Based Semantic Concept Indices

The text retrieval community has developed methods to extract certain kinds of semantic concepts, called named entities, from natural language text. **Named entity extraction** allows the categorisation of different types of objects in a sentence, for instance the names of organisations, locations, persons, expressions

of time, quantities, and so on. Being a statistical technique, named entity extraction can be trained to identify almost any category that occurs in the text, as long as there is sufficient tagged training data. Informedia uses named entity extraction to identify the names of people, organisations, and locations in ASR text, while NUS PRIS uses named entity extraction to identify people, organisations, and objects. They both create a separate index mapping different types of named entities to the shots that they occur in.

2.3 Request Analysis

The first task of any automatic search system when performing a new search is to analyse the user's request for information. The analysis is used to design queries for the different indices that have been outlined above, and it is sometimes also used to categorise different types of queries. All of the automatic search systems that have been reviewed incorporate two types of input: a textual information request, and a number of example videos that are relevant to the information need. Text offers important clues as to what concepts are present in a shot (Amir, Argillander et al. 2004; Yan, Hauptmann et al. 2004). Video can offer valuable clues about the low-level features that the desired shots contain, and can also be used to detect semantic concepts. We identify three different components of request analysis: index based analysis, where the input from the user request is decomposed into queries that are suitable for the available indices; meta-analysis and categorisation, where extra information is derived from the request through various techniques and used to categorise the request; and query formation, where the final queries that are to be sent to the indices are determined.

2.3.1 Index-Based Request Analysis

The same techniques used to create system indices are also used to create queries for those indices. By performing the same transformations on the request text and example videos as on the (ASR) text and video shots contained in the video collection, the defining characteristics of the request can be extracted. Different indexes can then be queried to find shots with similar characteristics. The methods used for index-based analysis usually correspond to the methods described in section 2.2. The systems use stopping, and sometimes stemming, on the request text to make it compatible for use with text indices. They also analyse the basic visual, auditory, and spatio-temporal characteristics of the example videos according to the low-level feature indices that are available. Example images are also used, but in this case only the visual features are used. IBM is the only team to perform video based semantic concept analysis on example video. NUS PRIS and Informedia, while performing named entity recognition on the request text, do not analyse the semantic content of the example video clips.

2.3.2 Request Meta-Analysis

IBM, DCU, and LowLands systems only perform simple index-based analysis on the textual requests that are passed to the systems. Informedia and NUS PRIS, however, perform further analysis that is not directly related to the indices contained in their automatic search systems. We term the type of analysis that is not immediately necessary for system indices, "request meta-analysis". NUS PRIS and Informedia use the information gained by meta-analysis to classify different types of requests, and to help determine the ways in which results should be integrated.

Request Meta-Analysis

An important form of meta-analysis is part-of-speech **tagging** (henceforth, "tagging") and **chunking** of the request text. A tagger assigns a grammatical part-of-speech classification, such as "plural noun" or "adjective", to every word in a sentence. (Charniak 1997). Different taggers employ different part of speech categorisations, but usually include distinctions between nouns, proper nouns, verbs, and adjectives. A chunker assigns grammatical classifications to groups of words at a phrasal level (Abney 1991). Chunkers usually distinguish between noun phrases, verb phrases, and prepositional phrases. Chunking and tagging algorithms are closely related, and both assign grammatical classifications to natural language text, only at different granularities. This is illustrated in Figure 5. Here we can see that part-of-speech tags are word-level units, while chunks are phrase-level units, and can be nested (for example, a noun chunk contained within a preposition chunk).

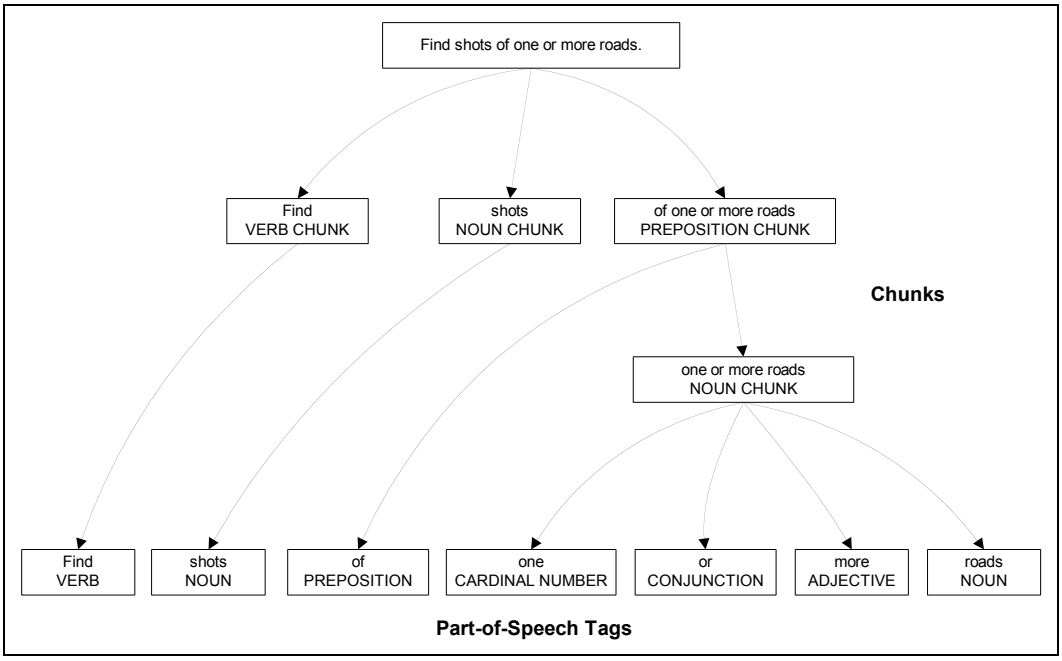


Figure 5. Parse tree of chunks and part-of-speech tags

Like named entity extractors, taggers and chunkers are generally based on statistical techniques. They require a large amount of manually tagged training data. Fortunately there are a number of pre-trained tagging and chunking systems available, such as the TreeTagger from Stuttgart University (Schmid 1994), and a number of manually tagged corpora such as the Penn-Treebank Corpus[8] have also been made public ally available. Manually tagged corpora can be used to train part-of-speech taggers and chunkers (Marcus, Santorini et al. 1993). Both Informedia and NUS PRIS emphasise the use of tagging and chunking to extract nouns and noun phrases as core terms for later queries.

There are two other types of meta-analysis which are done only by the NUS PRIS team. The first of these is "core term" detection. The NUS PRIS team has built a detector that identifies what they term "the strongest noun or noun phrase" in a sentence or collection of sentences. Though it is not explicitly stated, it appears that the core term detector identifies the subject of an information request. The second analysis done only by NUS PRIS is a look up of the words in the text in an online lexical reference source. Their system matches words in the text to words in the lexical reference system. This information is later used to retrieve synonyms and definitions of words that are then used to add to the queries for the text indices.

2.3.3 Request Categorisation

By categorising different types of information requests, NUS PRIS and Informedia hope to design different retrieval strategies that can be used in different types of situations. Both teams use the information obtained by named entity extraction and part-of-speech tagging to place an information request into one of several broad categories. Informedia distinguishes between different kinds of information requests: a named person; a named object; a general object; or a scene, defined as a request for multiple objects. NUS PRIS employs categorisations inspired by the different types of stories that are usually presented in news video: requests for a named person, for sports, for financial news, for weather related items, for disaster scenes, or for general stories that do not fit into any other categories.

Informedia and NUS PRIS rely on named entity extractors to identify requests for specific people. Named entity information is also used by Informedia to identify requests for specific named objects, while they employ chunking information to distinguish between requests for objects or scenes. Requests containing only one noun chunk are considered to be queries for general objects, while queries for multiple noun chunks are considered to be queries for multiple objects, and thus, scenes. NUS PRIS, on the other hand, uses word matching techniques to categorise requests that are not requests for named queries. They extract a list of keywords for sport, financial, weather, and disaster requests from a set of training examples. These keywords are then matched with words in the request and a category is determined.

[8] www.cis.upenn.edu/~treebank/

2.3.4 Relevance Feedback

Relevance feedback is a query expansion method that has been developed in the text retrieval field. Generally, blind relevance feedback methods operate by analysing the best results of an initial search and using the contents of those results to add terms to the original text query. In this way we can attempt to increase the number of salient terms we use in our search. NUS PRIS is the only team to incorporate relevance feedback. They use the top ten results of a text search to increase the number of query terms. They also search a large external corpus, namely the Internet, and extract terms from the top results of a query to the Google search engine[9], evaluating potential relevance of the new query terms by using mutual information (Church and Hanks 1990).

2.3.5 Query Formation

The essential function of request analysis is to form the queries that are to be passed to the system indices. To a large extent, this has already been accomplished in the index-based analysis. The query terms that result from this analysis can be directly sent to the various indices, to be fused later in the result integration (section 2.4). This is the approach taken by IBM and LowLands. Informedia and NUS PRIS further refine their queries by using query classification to help determine semantic concept queries. NUS PRIS also uses query expansion to find more terms for their text queries.

Informedia and NUS PRIS, like the other teams, make use of both low-level features and text from the information request to query the appropriate indices. Additionally, the request categorisation outlined in section 2.3.2 helps them to determine which semantic concept indices to query. They activate the face detector for queries for named people. In addition, NUS PRIS queries the concept detector that matches the request category if the request is for sport, finance, weather, or a disaster. Informedia uses part-of-speech information to reformulate the query for the text indices, searching only on noun chunks. The NUS PRIS team adds more terms to their text query by looking up of query terms in WordNet[10], an online lexical reference system that can be used as a dictionary. They add the synonyms and the descriptions of each of the matched words to search terms.

2.4 Result Fusion

Once the request has been analysed and queries have been passed to the appropriate indices, each query will return a result set. The fusion of results from different indices is an essential task when searching through a multimodal database, or any system that allows result sets from multiples indices to be combined into a single ranked list of results. In order to govern the fusion of different kinds of results, a weighting scheme is usually requisite.

2.4.1 Fusion Algorithms

Different types of indices typically have different types of relevance scores. It is not advisable to calculate the final relevance of a shot by adding the scores from all of the different indices together, as this will cause the results to be biased towards whichever index happens to give the highest relevance scores. Most teams have found it necessary to implement fusion algorithms that are independent of relevance score. Informedia, NUS PRIS and DCU have implemented a variation of the ranking-based Borda count called weighted Borda fusion. To implement the Borda count, the results from each index must be ranked so that the result with the highest relevance is ranked first, and the result with the lowest relevance is ranked last. Each result is given a score per index, where the ranking is equal to the number of shots that are less relevant than the current result. A very relevant result will have a high score. Finally, all of the scores are added together for each result. In weighted Borda fusion, a weight is assigned to the results from each index according to the level of confidence that that index will provide the correct results (Ho, Hull et al. 1994). LowLands and IBM do not use a rank based strategy. Instead, IBM normalises results from different sources so that they are not biased towards any one index. The weighting schemes used by LowLands provide results with similar types of weighting scores. They therefore combine results simply by adding the relevance scores from different indices, without requiring rank-based methods.

2.4.2 Weighting Schemes

Some indices are better indicators of how relevant a shot is than other indices, and different types of indices have different types of relevance weightings. The best possible relevance score for one type of index might be 0.5, while for another index the best possible score might be 500. This complicates the fusion of results. It is necessary to create weighting schemes to combine indices so that the best indicator of relevance has

[9] http://www.google.com
[10] http://wordnet.princeton.edu/

the largest contribution to the final score. Whatever the scheme, the creators of the reviewed systems all agree that up to this point in time, ASR has been the best source of information for searching.

Request-Independent Weighting Schemes

IBM, LowLands, and DCU employ the same weighting schemes for all information requests that they receive. They do not distinguish between different categories of information requests, and are therefore request independent. DCU incorporates a number of weighting schemes that were developed through optimisation with a set of development data. IBM incorporates a single weighting scheme which was also created through optimisation with a set of development data. All three found text to be the most important indicator of relevance, generally giving it a weight twice that of other types of indices.

Request-Dependent Weighting Schemes

NUS PRIS and Informedia adjust their weighting schemes according to different types of information requests. Informedia found that requests for named objects and named people, for example, benefited more from text than requests for general objects and scenes, as they are more likely to have a perfect match in ASR transcripts. Informedia uses two-step fusion architecture that they call mixture-of-expert architecture. In their model, text results are given very high importance, and results from other indices are only used to re-rank the results. NUS PRIS does not use a two-step approach, but also gives text a very high weighting. Both weighting schemes were optimised using a set of development data.

CHAPTER 3 THE AUTOSEEK SYSTEM

In the previous chapter we discussed the efforts of other researchers to realise a multimodal retrieval strategy for automatic video search. Now we will describe AutoSeek, a Java-based system that realises our strategy for automatic search with only a text request as input. Our strategy fits into the framework illustrated in Figure 6, which was developed through our review of other multimodal search systems. Our strategy is unusual in that it does not incorporate low-level feature indices, as we do not use multimodal examples to extract clues for low-level feature search.

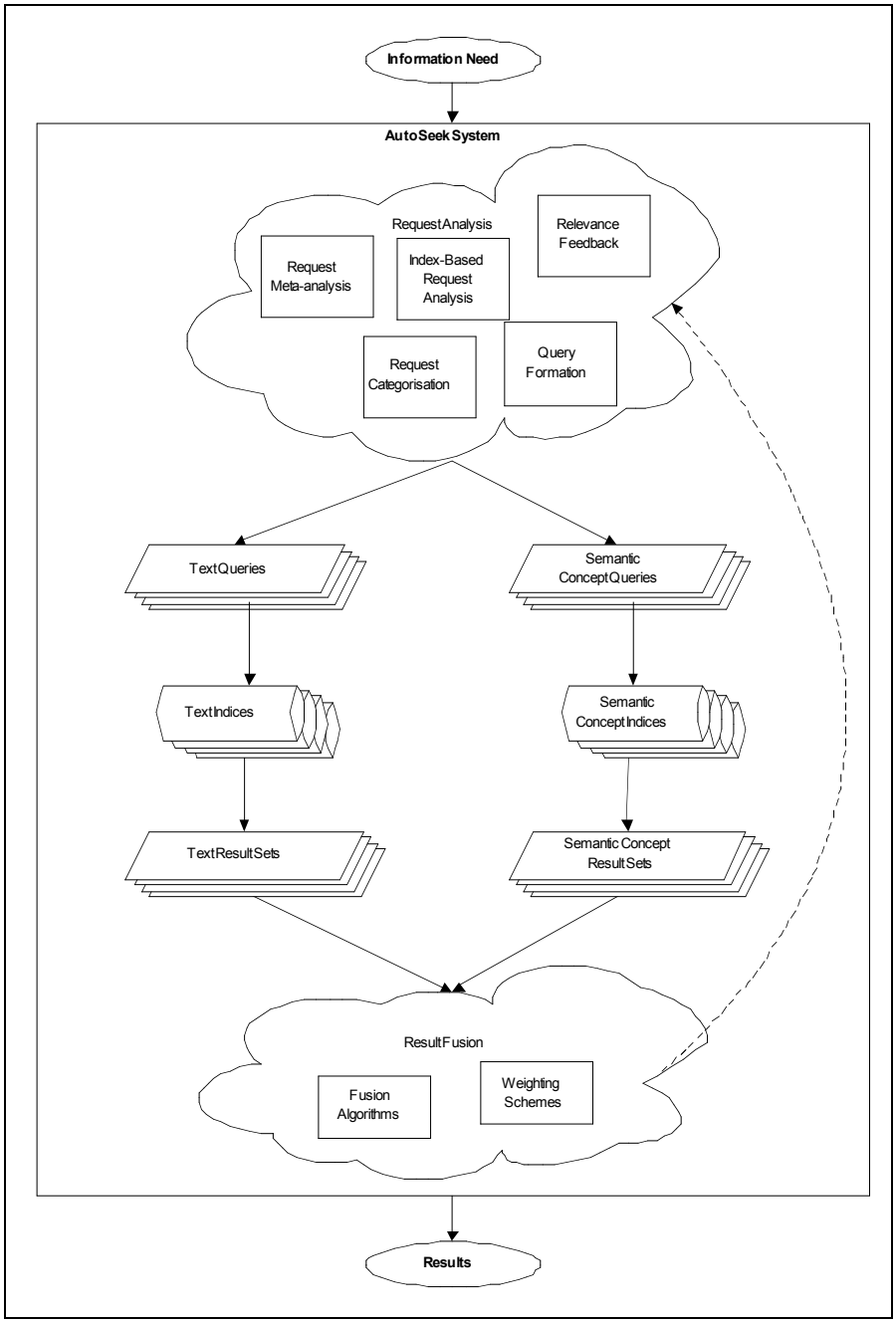


Figure 6. AutoSeek system design framework

We will start by outlining our development and test data sets, and follow this with an outline of the semantically-based strategy that we have developed. Finally, we outline the major components that have been implemented in AutoSeek.

3.1 Data Sets

AutoSeek has been designed, optimised, and tested using three large sets of news video data provided by TRECVID. Each data set is associated with one year of TRECVID evaluations. We use the data sets from 2003 and 2004 evaluations for development and optimisation of the AutoSeek system and the 2005 data for final testing. The video data for all years was accompanied by ASR speech transcripts generated using the LIMSI algorithm (Gauvain, Lamel et al. 2002), as well as a common definition for shot boundaries that is used to partition the video data into retrievable segments.

The data sets used for development and optimisation consisted of a total of approximately 130 hours of video footage captured from the CNN Headline News and ABC World News Tonight channels in 1998. They were accompanied by 46 "topics"[11], statements of information need that include a natural language request (for example, "Find shots of Saddam Hussein") and a number of multimodal examples. We only utilise the request for our system. The **ground truth** for each question was also available. The ground truth identifies the shots in the data set that contain the requested content.

The data set that was used for evaluation contained approximately 80 hours of video footage captured from the Arabic LBC, the Chinese CCTV4 and NTDTV, and the American CNN and NBC channels in the last half of 2004. Off-the-shelf commercial machine translation was used to provide ASR information for the non-English channels.

3.2 The Semantic Challenge

We use semantic concepts as the basis of our solution for translating text into multimodal queries. By identifying a large number of semantic concepts in the video collection and subsequently matching these to any concepts that are found in the text of information requests, we will be able to identify which multimodal concept indices should be used for each request. AutoSeek needs to analyse raw video data and identify any semantic concepts that it contains. It must analyse textual data and identify any semantic concepts there. Each of these steps requires machine translation form raw digital data to a human interpretation of the same data. Finally, AutoSeek must recognise relationships between different semantic concepts.

3.2.1 Detecting Semantic Concepts in Video

A large number of specialised algorithms have been developed for the detection of semantic concepts in video data, resulting in a large number of isolated and specialised concept detectors. Although specialised detectors have been invaluable in achieving progress in the semantic concept detection domain, they must be superseded by more generic detection methods. Only through generic methods will we be able to create detectors for the huge numbers of semantic concepts that users may wish to search for in video. To this end, researchers at MediaMill have developed a generic approach for concept detection, the Semantic Pathfinder (Snoek, Worring et al. 2005). We give a brief overview of the Semantic Pathfinding strategy in this section.

<u>The Video Authoring Process</u>

The strategy of the Semantic Pathfinder is based on the video authoring process. The different steps taken by a cinematographer creating video guide the Semantic Pathfinder in detecting different types of semantic concepts. The Semantic Pathfinder identifies consecutive steps in the video authoring process: first context, followed by style, and finally content. The authoring process is illustrated in Figure 7 with an example. This video has been authored within the context of an overseas news report, the context of a war story, and the context of embedded reporting. Some of the style decisions made when filming the shot were to record a mid-shot of the reporter, to overlay a caption detailing the name and location of the reporter, and to film a steady shot with no camera movement. Finally, some of the content of the shot includes a person, along with skin and clothing. There are flames and smoke in the background.

[11] Originally 49 requests were provided, but 3 requests could not be incorporated as our data set did not contain any relevant shots.

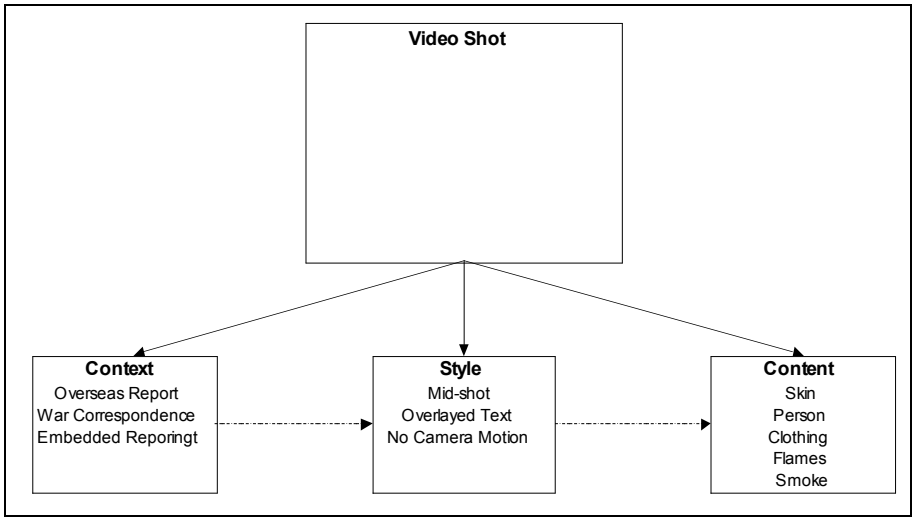


Figure 7. The authoring of a video shot

The Semantic Pathfinder reverses the authoring process. The strategy is to analyse the video using a series of links, first analysing the content of a shot, then its style, and finally its context, as is shown in Figure 8. Validation is then used to select the link that is the best indicator of concept presence. The probability output of each link can be used as input for the next link. Other forms of input for the links are raw multimedia video data and also video annotation data. Raw multimedia data can be used to extract low visual, auditory, and spatio-temporal characteristics of the video data. Video annotation data can be used to identify speech characteristics of data. It also includes the output of a number of specialised detectors. The strength of the Semantic Pathfinder is that it uses the different elements of the authoring process to help select the right path to take in concept detection, allowing it to take into account elements such as the style of video, which are not exploited by other generic detection methods.

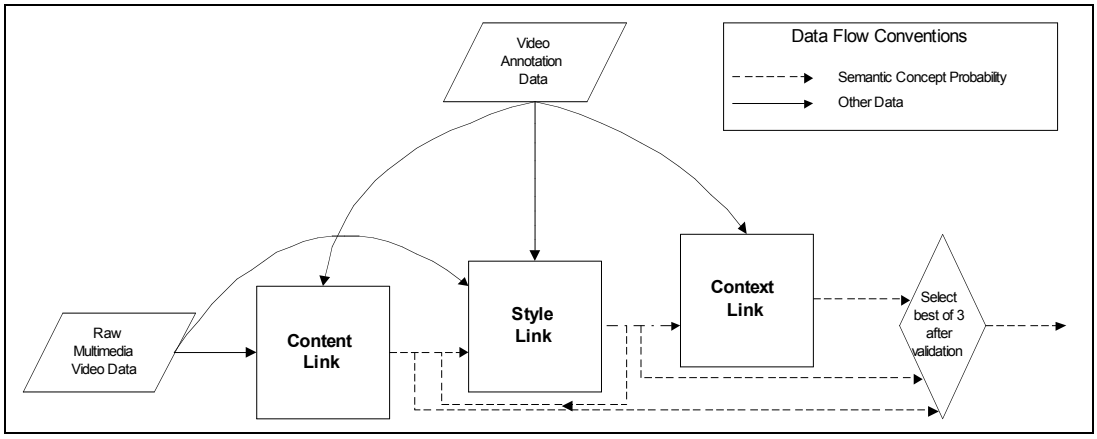


Figure 8 Semantic Pathfinding for one concept, adapted from Snoek (2005)

3.2.2 Detecting Semantic Concepts in Natural Language Text

While digital video is a relatively new medium and research into video analysis is relatively young, natural language text has been present in the digital world ever since the first word processors, text editors, and communication programs were created. Many resources have been developed for the processing of natural language text, ranging from automatic machine translation between languages, to text advances retrieval algorithms, to grammatical analysis. There are two general types of approach for solving natural language problems: statistically-based methods and linguistically-based methods. Statistically-based methods rely on techniques that treat text as a "bag of words", while linguistically inspired approaches rely on knowledge bases and formal semantics. Statistically-based methods have usually been found to work best, as by using large amounts of data they are able to implicitly capture features of text that have to be explicitly defined when using linguistically-based methods (Voorhees 1999). AutoSeek therefore incorporates statistically-based techniques whenever possible, but for the detection of semantic concept indices in text it must take a linguistic approach.

We only have a set of 46 sample requests that can be used for development, a number too small to enable a statistical approach.

WordNet: A Semantic Knowledge Database

AutoSeek detects semantic concepts by translating terms in the information request to terms in a large lexical database. The database that AutoSeek uses should fulfil a number of criteria. It must have explicit relational connections between the different concepts that it contains, so that concepts detected in the text can be related to the available semantic concept indices. It should encompass a very large number of different semantic concepts, as it is impossible to predict beforehand what kind of concepts users will search for. It should be available as open-source so that it can be incorporated into the AutoSeek system. Given these criteria, WordNet[12] is the obvious choice for use in AutoSeek. WordNet is a publicly available lexical reference system that was developed by the Cognitive Science Laboratory at Princeton University. It is updated regularly, and the latest release defines over 117,000 different semantic concepts[13] which are connected through different types of explicit relationships. A number of interfaces and toolkits for WordNet are available; we use the Java WordNet Library[14].

WordNet is a large database consisting of many semantic concepts, or **synsets.** We will use the words "synset" and "concept" interchangeably within the context of WordNet. Each synset is described by one or more words, and a definition that describes the semantic concept it is associated with. Each synset is also defined by its grammatical categorisation as a noun, verb, or adjective (other word types are not included in WordNet). We illustrate this in Figure 9 with a depiction of the synsets associated with the word "person" in WordNet. We see that the word "person" is associated with three different synsets, all of them nouns. The word "person" can mean a human being, but it can also used as a reference to the human body, or as a grammatical category.

Grammatical Type	Synset Words	Synset Definition
Noun	person, individual, someone, somebody, mortal, soul	a human being. "there was too much for one person to do"
Noun	person	a human body (u sually including the clothing). "a weapon was hidden on his person"
Noun	person	a grammatical category of pronouns and verb forms. "stop talking about yourself in the third person"

Figure 9. Synsets associated with the word "person"

A number of relationships between synsets have been encoded in WordNet, of which only four were considered for AutoSeek. They are both transitive relationship pairs, consisting of the **hyponym** and **hypernym** pair and the **meronym** and **holonym** pair, illustrated in Figure 10. The first relationship pair describes the specificity of different concepts, and is often described as the "is-a" relationship. A concept is a hyponym of another concept when it is a more specific type of that concept. The hypernym relationship is the inverse of the hyponym relationship: a concept is a hypernym of another concept when it is a more general type of that concept. The meronym and holonym relationships are often described as the "part-of" relationships. A concept is a meronym of a second concept when it is a constituent part of that concept, while inversely it is a holonym of a second concept when the second concept is a part of that concept.

Consider the illustrated example of this relationship. The concept "person, individual, someone…" is more specific than the concept "organism, being", and is therefore a hyponym of that concept. "Person, individual, someone…" is a more general type of the concepts "female, female person" and "male, male person", and is therefore a hypernym of those concepts. The second relationship pair describes the "part-of" relationship. Thus we can see that the concepts "face, human face", "hand, manus, mitt, paw", "arm", and "foot" are all parts of the concept of a "homo, man, human being, human", and are therefore its meronyms. Inversely, the concept "homo, man, human being, human" is a part of the concept representing the class of mammals, "Mammalia, class Mammalia", and is therefore a member holonym of that concept.

[12] http://wordnet.princeton.edu/
[13] Version 2.1 statistics, http://wordnet.princeton.edu/man/wnstats.7WN
[14] http://sourceforge.net/projects/jwordnet

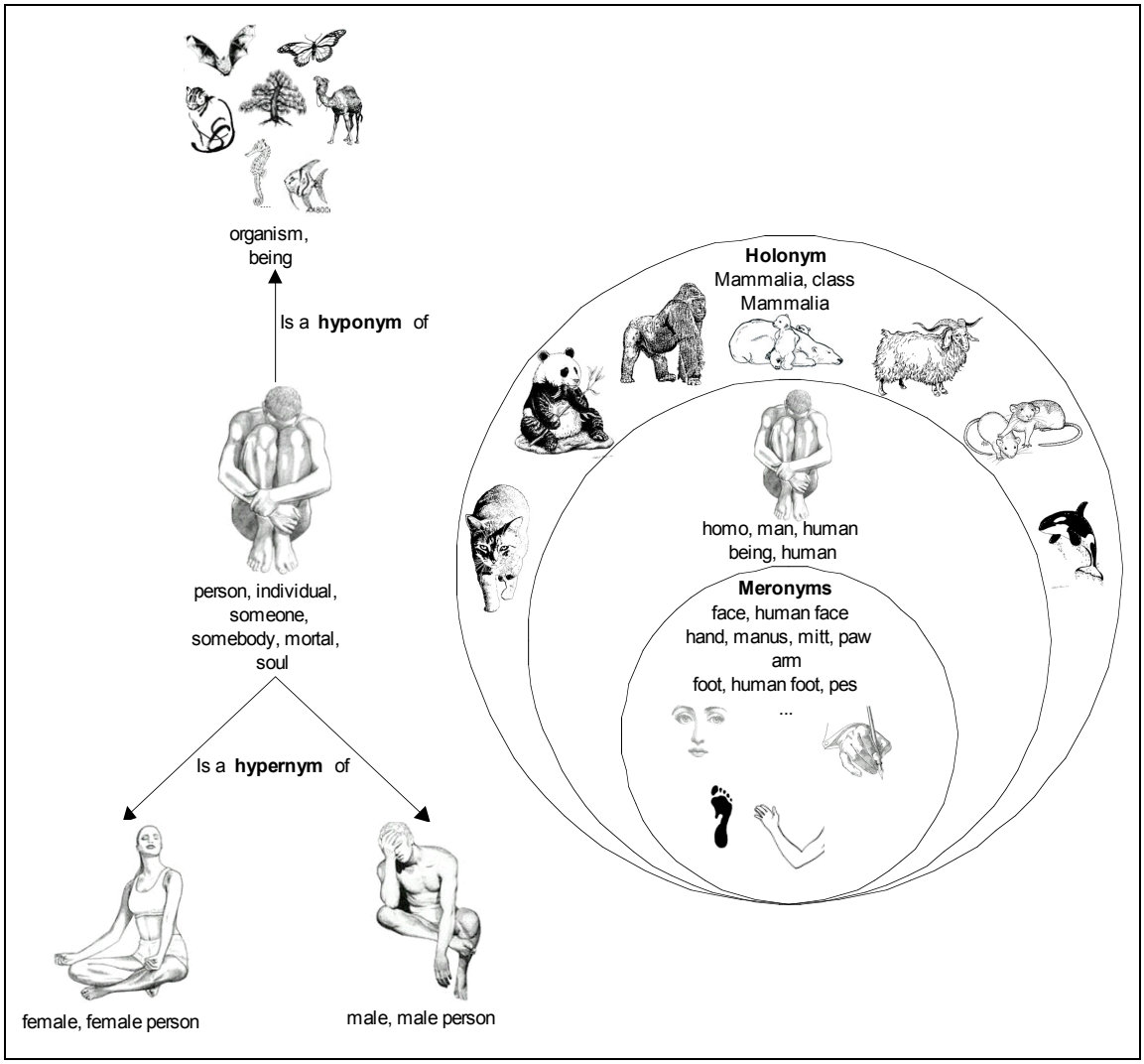


Figure 10. An example of the transitive hyponym/hypernym and holonym/meronym relationships[15]

Extracting WordNet Concepts from Text

The first step that we take in extracting WordNet concepts from text is to identify which words in the text are nouns. We subsequently look up each noun in WordNet. Once the WordNet nouns have been identified, we still need to identify the correct meaning for each noun that is associated with multiple interpretations. Approximately 17% of the nouns in WordNet have more than one meaning, making this no trivial matter. For example, in the sentence "A basketball goes through a hoop", it is necessary to determine whether we are talking about basketball as a kind of ball, or about basketball as a kind of game. To determine which meaning we are talking about, it is necessary to consider the context of the word basketball. For a human observer it is relatively simple to determine that a game cannot pass through a hoop, while a ball can. For an automatic system, this is not so simple, and we must perform **disambiguation** to determine which meaning of the word is correct. Much research has been done into the disambiguation of nouns in WordNet (Resnik 1995; Jiang and Conrath 1997; Ide and Véronis 1998; Leacock and Chodorow 1998; Lin and Shavlik 1998; Pedersen, Patwardhan et al. 2004). Automatic disambiguation of words with multiple senses in general relies on two major information sources; the *context* of the word that should be disambiguated, and *external knowledge sources* that provide extra information that can be used to help determine the correct sense of a word (Ide and Véronis 1998). In the AutoSeek system, the context that is used for disambiguation is all of the WordNet nouns

[15] Relationships between concepts in this diagram have been simplified from actual WordNet relationship for the purposes of this example

that are contained in an information request. The external knowledge source is the WordNet lexicon. To find which sense of each word should be used, the most interrelated senses are selected.

The final disambiguation algorithm, based on the algorithm outlined by Resnik (1995) is shown in Figure 11. We calculate the similarity between possible meanings of different words. The sense of each word that has the highest similarity is selected as the correct sense. If more than one sense has the same similarity score, then the most common sense of the word is selected as the best sense.

For each WordNet noun, *a*

For each meaning, $sense_i(a)$

For each WordNet noun, *b,* where $b \neq a$

For each meaning, $sense_j(b)$

Calculate similarity *s* between $sense_i(a)$ and $sense_j(b)$

Increment the support for $sense_i(a)$ and $sense_j(b)$ by *s*

Figure 11. Similarity-based disambiguation algorithm for WordNet nouns

3.3 Indices

3.3.1 Text Indices

TFIDF Indices

AutoSeek contains two indices for text search that have been developed according to the traditional TFIDF (Term Frequency Inverse Document Frequency) model described in section 2.2.2. One index contains stemmed terms, and the other contains unstemmed terms. This allows us to compare the effects of searching on unstemmed text as opposed to stemmed text. The indices map each term that occurs in the video collection to all of the shots that contain that term in the ASR. In the literature review we discussed video specific augmentations for text search, such as retrieving shots temporally adjacent to the shot containing the query term. We have not implemented these types of strategies in AutoSeek due to time restrictions.

After experimenting with a number of different weighting schemes, we adopted the classical TFIDF weighting scheme *bfx·bfx* (Salton and Buckley 1988) for text retrieval. This scheme is characterised by a lack of normalisation for document length, and the employment of a boolean term frequency of 1 or 0. This scheme works well because the ASR transcripts for the video shots do not vary enormously in length, thereby making normalisation for document length less important. Also, when a term is mentioned several times in a shot transcript, this does not necessarily make it less likely that the required visual object will contained in the shot. The final relevance per shot is calculated by:

$$\sum_{i \in q \cap d} (1 \cdot \log \frac{N}{n_i}) \cdot (1 \cdot \log \frac{N}{n_i}) \tag{3.1}$$

where q is the collection of terms from the original query, d is the collection of terms in the ASR text document associated with a particular shot, N is the total number of shots in the collection, n is the total number of shots containing the current query term, and i is a query term.

The SMART stop list (Salton 1971) of 524 common English words was used to filter out stop words from the ASR during the indexing procedure. The Porter stemming algorithm (Porter 1980) was used to stem the words from the ASR when creating the stemmed index.

LSI

AutoSeek incorporates **Latent Semantic Indexing** (LSI) as defined by Deerwester, Dumais et al. (1990). LSI was implemented by MediaMill for systems taking part in previous TRECVID evaluations (Worring, Nguyen et al. 2003; Snoek, Worring et al. 2004), and is a search method for text that uses implicit associations between clusters of terms to automatically expand a query beyond the original terms. The LSI index is generated by first creating a high-dimensional vector space consisting of all the terms[16] contained in the ASR transcript, and then reducing the number of dimensions using principal component analysis. When a query is made to the LSI index, the terms of the query are placed in the reduced dimensional space. The shots that are

[16] Excluding stop words.

closest to the query in that space are returned, regardless of whether they contain the original query terms. In this way, for example, a query using the terms “saddam” and “hussein” might return a shot containing the term “iraq” even if it contains neither the term “saddam” nor the term “hussein". The main advantage of using LSI for AutoSeek is that it can be used to return any number of shots from the video collection, unlike TFIDF indices which only return the shots that contain the original query terms. This is important within the evaluation framework, where the first 1,000 ranked shots are evaluated.

3.3.2 Semantic Concept Indices

The AutoSeek system incorporates a number of semantic concept indexes that were generated by MediaMill at the University of Amsterdam. Each index relates to a high-level, real world concept. These concepts represent a mixture of people, things, events, and locations. These indices were chosen from different semantic concept indices that MediaMill had created for the data development and evaluation sets for previous experiments. There were 32 indices available for the development video data, and there were 101 indices available for the test video data. An overview of these indices is given in Figure 12.

Semantic Concept Indices - Development Set (2003 and 2004 data sets)						
aircraft	basketball	car	football	outdoor	soccer	*violence*
Albright	beach	cartoon	golf	overlaid text	sport	weather
anchor	bicycle	Clinton	*graphics*	*people*	studio	
animal	boat	Dow Jones	ice hockey	*people walking*	train	
baseball	building	financial	monologue	road	vegetation	

Semantic Concept Indices - Test Set (2005 data set)						
aircraft	candle	*duo anchor*	*grass*	mountain	road	tower
Allawi	car	*entertainment*	Nasrallah	natural disaster	screen	tree
anchor	cartoon	explosion	horse	newspaper	Sharon	truck
animal	chair	*face*	horse racing	*night fire*	sky	urban
Arafat	*charts*	*female*	house	office	smoke	vegetation
baseball	Clinton	fire weapon	Hu Jintao	outdoor	snow	vehicle
basketball	cloud	fish	*indoor*	*overlaid text*	soccer	*violence*
beach	corporate leader	flag	Kerry	*people*	*split screen*	walking/running
bicycle	court	flag USA	Lahoud	people marching	sports	water body
bird	crowd	food	*male*	police security	studio	waterfall
boat	cycling	football	maps	Powell	swimming pool	weather
building	desert	golf	meeting	prisoner	table	
bus	dog	government building	military	racing	tank	
Bush Jr.	drawing	government leader	monologue	religious leader	tennis	
Bush Sr.	drawing/cartoon	*graphics*	motorbike	river	Tony Blair	

Figure 12. Semantic concept indices created for the development and test sets (those not used in AutoSeek are shown in italics)

Each semantic concept index is linked to one or more concepts in WordNet. The linked concepts in WordNet are selected on an index-by-index basis, according to whether they belong to the concept or are an integral part of the concept. For example, the detector for the concept “baseball” finds shots of baseball games, and these shots invariably include baseball players, baseball equipment, and a baseball diamond. These concepts are holonyms of the concept baseball, in that together they make up a baseball game, but the meronym/holonym relationships in WordNet are not encoded completely enough that they will always be recognised as being related. Extra functionality has been added to AutoSeek to enable named objects that are not contained in WordNet (such as “Madeline Albright”) to be included.

Not all of the available indices were actually incorporated in the AutoSeek system, as shown by the italicised concept names. Of the 32 concept indices available for the development set, and the 101 concept indices available for the test set, we use only 28 and 88 indices respectively. To select indices for AutoSeek we made use of three rules-of-thumb:

1. The semantic concept index should not represent a very common concept, one that occurs in a large number of video shots.
2. The semantic concept index should be trained using examples that are representative of all instances of that concept in the data set.
3. The semantic concept index should not represent a layout element of a shot.

The motivation for the first rule is the same as the motivation for removing stop words from text indices: commonly occurring items in the search set have little discriminatory value. AutoSeek selects only the most similar index in a search. If that index assigns a high relevance to most of the shots in the collection, then it will be of little added value. Therefore, for example, the index for the semantic concept "people" is not used. The second rule is introduced because some detectors are trained using sub-segments of the data-set that are not representative of the concept as it occurs as a whole. For example, the detector used to create the "violence" indices almost exclusively uses shots containing sports and fires as training examples. This training strategy creates a detector that is very successful in other TRECVID tasks, but is not very useful for the AutoSeek search task. A user searching for "street violence" is unlikely to want to receive shots of soccer matches as a result. The motivation for the last rule lies in the implementation of WordNet in AutoSeek. Layout elements such as "split screen" are not specifically encoded in WordNet, and an entirely new relationship tree would have to be encoded to accommodate these specialised concepts. This was not possible in AutoSeek; therefore these indices were not included.

3.4 Request Analysis

3.4.1 Index-Based Request Analysis

The query text undergoes simple formatting and analysis in the form of stop word removal and stemming to make it compatible with the text indices. In addition, punctuation is removed and all letters are converted to lower case. Two domain-specific stop words are added to the stop list: "find" and "shots". These two words occur in every information request contained in the development data, and therefore have no value for any specific search. Example 1 shows the results of formatting and stopping input text.

Input: Find shots of a graphic of Dow Jones Industrial Average showing a rise for one day.
Output: graphic dow jones industrial average showing rise day

Example 1. Results of formatting and stopping a sentence

After stopping has been done a stemmed version of the stopped text is created using the Porter stemming algorithm, previously discussed in 2.2.2. This results in the removal of common suffixes, as can be seen in Example 2.

Input: graphic dow jones industrial average showing rise day
Output: graphic dow jone industri averag show rise day

Example 2. Results of stemming text that has been stopped

3.4.2 Request Meta-analysis

Grammatical analysis

The AutoSeek system generates two different types of grammatical information about a natural language information request: the part-of-speech assignment of each individual word, and categorisations of types of chunks. The grammatical information is generated using the TreeTagger (Schmid 1994) part-of-speech tagger, which was chosen because it has a high level of accuracy (96.36% on Penn Treebank data), as well as being a publicly available and "ready-to-go" tool that does not require any specific training.

The type of information produced by analysing the parts-of-speech of a topic is shown in Example 3. Each word is assigned a label by the TreeTagger indicating its grammatical classification. Each word occurs only once, and there are many different grammatical classifications (a full description of the various labels can be found in Appendix II).

Input: Find shots of a graphic of Dow Jones Industrial Average showing a rise for one day.

Output:

Text	Label	Part-of-speech	Text	Label	Part-of-speech
Find	VV	Verb	Average	NP	Proper noun
shots	NNS	Plural noun	showing	VVG	Verb, present participle
of	IN	Preposition	a	DT	Determiner
a	DT	Determiner	rise	NN	Noun
graphic	JJ	Adjective	for	IN	Preposition
of	IN	Preposition	one	CD	Cardinal number
Dow	NP	Proper noun	day	NN	Noun
Jones	NP	Proper noun	.	SENT	End of a sentence
Industrial	NP	Proper noun			

Example 3. Results of part-of-speech analysis of a sentence

Example 4 shows the results of a chunk analysis of a sentence. Here we can see words are analysed at a phrasal level. The same word may occur in multiple chunks, but no chunk has precisely the same content. There are three types of chunks: verb chunks, noun chunks, and preposition chunks. These chunk types indicate the overall syntactic function of the phrase that they are assigned to. Thus, the group of words contained by the phrase "of Dow Jones Industrial Average" functions as a preposition, while the group of words contained in the phrase "Dow Jones Industrial Average" functions as a noun.

Input: Find shots of a graphic of Dow Jones Industrial Average showing a rise for one day.

Output:

Text	Label	Chunk Type
Find	VC	Verb chunk
shots	NC	Noun chunk
of a graphic	PC	Preposition chunk
of Dow Jones Industrial Average	PC	Preposition chunk
Dow Jones Industrial Average	NC	Noun chunk
showing	VC	Verb chunk
a rise	NC	Noun chunk
for one day	PC	Preposition chunk
one day	NC	Noun chunk

Example 4. Results of chunk analysis of a sentence

Semantic Concept Extraction

As outlined in section 3.2.2, we use WordNet to identify different types of semantic concepts in text. We are only interested in concepts that are identified as nouns because all of our semantic concept indices describe object. We determine which concepts are present as nouns with the grammatical information generated by the TreeTagger. To extract the candidate semantic concepts, AutoSeek first analyses each noun chunk to see if it contains a WordNet noun comprises more than one word. After this has been done, each noun for which a WordNet noun has not yet been retrieved is looked up in the WordNet dictionary. Example 5 illustrates the results of the WordNet noun extraction algorithm for a set of noun chunks and nouns.

Input (text)	Label	Description	**Output (WordNet Noun)**
Dow Jones Industrial Average	NC	Noun chunk	Dow Jones
a rise	NC	Noun chunk	-
one day	NC	Noun chunk	-
Dow	NP	Proper noun	-
Jones	NP	Proper noun	-
Industrial	NP	Proper noun	-
Average	NP	Proper noun	average
Rise	NN	Noun	rise
Day	NN	Noun	day

Example 5. Results of WordNet analysis for a set of noun chunks and nouns

In section 3.2.2 we discussed the problems associated with words that have multiple meanings, and an algorithm for finding the correct meaning by using the similarity to surrounding words. We use this algorithm within AutoSeek, implementing six different similarity measures for WordNet nouns with the Perl module WordNet::Similarity (Pedersen, Patwardhan et al. 2004).

Three of the similarity measures are based on information content, an idea adopted from information theory and first applied to WordNet by Resnik (1995). Within his definition of information content, the rarer a concept is used, the more informative it is. Information content is calculated by counting the occurrences of a concept and all of its hyponyms in a large tagged corpus, and transforming this count by a log function so that information content decreases as the number of occurrences decreases. Resnik defines the similarity between two concepts A and B to be the same as the information content of the most informative common ancestor shared by both concepts. Jiang and Conrath (1997) and Lin and Shavlik (1998) augment Resnik's measure with the sum of the information content of concepts A and B. Jiang and Conrath take the difference of this sum and the information content of the most informative common ancestor, while Lin and Shavlik scale the information content of the ancestor using this sum.

Two measures are based on the number of connections between two concepts. Leacock and Chodorow (1998) and Wu and Palmer (1994) both use the shortest number of jumps between concepts A and B using the hyponym and hypernym relationships. Leacock and Chodorow scale the number of jumps by the maximum possible number of jumps in the entire relationship tree to come to a similarity score, while Wu and Palmer use concept depth in the entire hierarchy to scale the number of jumps and come to a similarity score. The final measure uses implicit as well as explicit relationships to calculate the similarity between two concepts. Banerjee and Pedersen (2003) also incorporate statistical information about the amount of overlap of the WordNet definitions of concepts A and B and adjacent concepts to calculate how closely they are related.

The similarity measures described sometimes assign the same similarity score to multiple concepts. This is problematic in disambiguation, where the similarity score is used to choose the best meaning of a word. WordNet ranks different meanings of the same word according to how common each meaning is. If the similarity information is insufficient to disambiguate, we select the most common meaning of a word. This information is also used as a final test of the disambiguation measures. We define an additional disambiguation algorithm which simply chooses the best meaning of a word. In Example 6 we illustrate the input and output for ideal disambiguation of a topic. At this stage we do not yet choose a single disambiguation measure to use for AutoSeek, instead we perform optimisation experiments described in the next chapter to determine the best approach.

Input:

WordNet Noun	Candidate Meaning
Dow Jones	an indicator of stock market prices; based on the share values of 30 blue-chip stocks listed on the New York Stock Exchange
Average	a statistic describing the location of a distribution
rise	a growth in strength or number or importance
	the act of changing location in an upward direction
	an upward slope or grade (as in a road)
	the property possessed by a slope or surface that rises
	(theology) the origination of the Holy Spirit at Pentecost
day	time for Earth to make a complete rotation on its axis
	some point or period in time
	the time after sunrise and before sunset while it is light outside
	a day assigned to a particular purpose or observance
	the recurring hours when you are not sleeping (especially those when you are working)
	an era of existence or influence
	the period of time taken by a particular planet (e.g. Mars) to make a complete rotation on its axis

Output:

WordNet Noun	Disambiguated Meaning
Dow Jones	an indicator of stock market prices; based on the share values of 30 blue-chip stocks listed on the New York Stock Exchange
Average	a statistic describing the location of a distribution
rise	a growth in strength or number or importance
day	time for Earth to make a complete rotation on its axis

Example 6. Results of a disambiguation implementation in AutoSeek

3.4.3 Request Categorisation

We two categorisations of requests: specificity and complexity. The algorithms for both categorisations are summarised in Figure 13. We derive the specificity categorisation from Hollink, Nguyen et al. (2004), who analysed different user classifications of information requests provided in the TRECVID evaluations. They identify the categorisation of **specific** vs. **general** requests to be an important distinction between different requests. We implemented this categorisation in AutoSeek through grammatical meta-information about each information request. If a request contains a proper noun it refers to a specific object, rather than a general category, so AutoSeek categorises all requests containing proper nouns as specific requests, and all others as general requests. "Find shots of last year's tsunami hitting Thailand", for example, is referring to a specific occurrence in news video. It contains a proper noun, "Thailand", and will therefore be correctly classified by AutoSeek as a specific topic.

Specificity Categorisation

Request contains proper noun?

YES → SPECIFIC

NO → GENERAL

Complexity Categorisation

Request contains more than one noun chunk?

YES → COMPLEX

NO → SIMPLE

Figure 13. Algorithms for request categorisation

Another categorisation that is implemented in AutoSeek is that of **simple** vs. **complex** requests. We use this categorisation to distinguish between requests for a single object (simple requests), and requests involving multiple objects (complex requests). Our definition of complex requests is similar to Informedia's definition of scenes as described in 2.3.3, differing in that requests containing a named person or a named object can also be categorised as complex requests. Here we also use grammatical meta-information to perform categorisation. Any request containing more than one noun chunk is classified as complex, as it refers to more than one object, while requests containing only a single noun chunk are classified as simple. The example used in the previous paragraph would be classified as a complex query, as it contains two noun chunks: "last year's tsunami" and "Thailand".

3.4.4 Query Formation

The strategy that is used for query formation has been determined by optimisation using development data, and will therefore be discussed in the next chapter.

3.4.5 Blind Relevance Feedback

AutoSeek incorporates relevance feedback. We have used it primarily in our preliminary investigation of the best kinds of text queries to use for different requests, outlined in section 4.2. AutoSeek is also capable of blind relevance feedback, which can be used to expand queries even when the correct results are not known. This is done simply by assuming that the top ranked results of a search are correct.

We employ Rocchio's method for relevance feedback, a vector-based algorithm that has been shown to work well in the past (Salton and Buckley 1990). The formula for Rocchio's relevance feedback is

$$Q_{new} = Q_{old} + \beta \sum_{\substack{n_1 \text{ rel} \\ \text{docs}}} \frac{D_i}{n_1} - \gamma \sum_{\substack{n_2 \\ \text{non-rel} \\ \text{docs}}}^{n_2} \frac{D_i}{n_2} \tag{3.3}$$

A new query vector, Q_{new} is created by analysing examples of n_1 relevant and n_2 irrelevant documents. Each unique term in each document vector, D, is assigned a weight according to the term frequency and the inverse document frequency. The positive values are weighted using a constant, β, and the negative values are weighted by another constant, γ. These constants are set at 0.75 and 0.25 respectively, as recommended by Salton and Buckley (1990). Finally, a new weight is calculated for every term in the collection by adding the weightings for positive examples and subtracting the weightings for negative examples. The terms with the highest value are considered the most relevant to the query. To adjust the relevance feedback formula for blind relevance feedback, we select the top results of a search as positive examples, and do not provide the system with any negative examples.

3.5 Result Fusion

3.5.1 Fusion Algorithms

AutoSeek implements the rank-based weighted Borda fusion described in section 2.4.1. AutoSeek adopts the weighted Borda fusion algorithm described in (Ho, Hull et al. 1994), substituting shots instead of classifiers as input to be ranked. To perform Borda fusion, first each result set must be assigned a weight and the Borda count for each shot, x, must be calculated for each result set. We discuss the development of weighting schemes in section 4.3.2. The Borda count for a shot can be calculated by

$$N_i - r_i \tag{3.4}$$

where N_i is the number of results in a result set, i; and r_i is the ranking of the current result in the set of results from index i, where the highest ranking result is given a weighting of 0 and the worst result is given a ranking of N-1. The Borda count will therefore give the highest ranked shot a score equal to the number of results, N, and the lowest ranked shot is given a score of 1.

The final relevance of a shot, x, after combining results from different indices is given by

$$\alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m \tag{3.4}$$

where $x = (x_1, x_2, \ldots, x_m)$ represents the Borda count assigned to the current shot in each result set $R_1, R_2, \ldots, R_m$; α represents a constant parameter; and $\beta = (\beta_1, \beta_2, \ldots, \beta_m)$ represents the weights assigned to each of the result sets.

One alteration was made to the generic algorithm described above. The Borda algorithm was designed for working with sets of equal size, i.e. where N is equal for all sets. However, in AutoSeek some result sets may be very small while others may be very large. This can skew weightings. For example, if one result set contains 18 results, and another 2,000, the top ranked result in the first set will be given a Borda count score of 18, and the top result of the second set will be given a Borda count score of 2,000. To compensate for this, N is set to a default value of 1,000 for all result sets. This value for N was chosen because in the context of the TRECVID evaluation AutoSeek is required to deliver a final result set of 1,000 results.

3.5.2 Weighting Scheme

The strategy that is used for weighting schemes is determined by optimisation using development data, and will therefore be discussed in the next chapter.

CHAPTER 4 SYSTEM OPTIMISATION

In the previous chapter we outlined the design and implementation of the AutoSeek system, which is summarised in Figure 4. We left the implementation of system components open, and these are shown in bold face in the diagram: the query formation component, and the final fusion strategy component. These components are very much dependent on the data set that is being employed, and are the focus of our optimisation experiments.

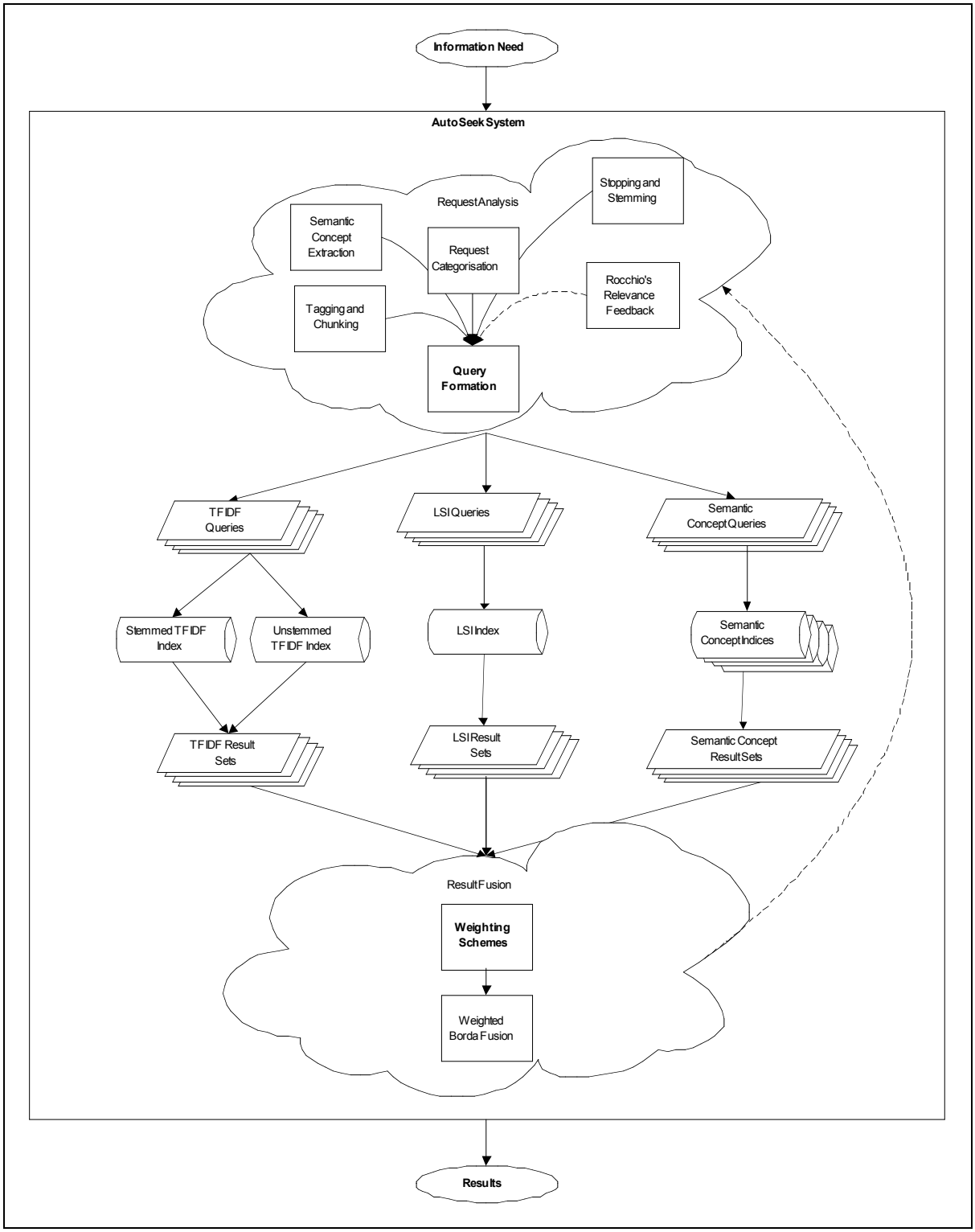


Figure 14. The components and structure of the AutoSeek system

4.1 Evaluation Methodology

Evaluation Measure

The primary evaluation measure that has been applied throughout our experimentation is non-interpolated average precision, the standard evaluation measure of the TRECVID evaluations. Average precision is a combination of two common information retrieval evaluation measures, precision and recall. Precision measures the ability of a system to retrieve only relevant items from a set of items. Recall measures the number of relevant items that have been retrieved from a set of items. If a set of results contains many irrelevant items, the precision will be low. If a set of results does not contain many of the available relevant items, the recall will be low. The formulae for precision and recall (Voorhees and Harman 2001) are:

$$recall = \frac{r}{R} \quad (4.1) \qquad precision = \frac{r}{N} \quad (4.2)$$

where r is number of retrieved relevant results, N is total number of retrieved results, and R is number of relevant items in collection.

Average precision combines recall and precision in one measure by calculating the precision at every point at which a correct result is returned (in other words, at every point that recall increases), and averaging this for all of the correct items in the total collection. The formula for average precision at a given rank k for an answer set A is:

$$average\ precision = \frac{1}{R}\sum_{k=1}^{A}\frac{R \cap N^k}{k}\lambda(n_k) \quad (4.3)$$

where N^k represents f, a ranked version of A, and where the indicator function $\lambda(n_k) = 1$ if $n_k \in R$ and 0 otherwise (Snoek, Worring et al. 2005).

Average precision rewards returning correct items at the highest ranking, as can be seen in Figure 15. This examples shows an average precision curve for a set of 20 items, with only one relevant item which is shown at different ranks. Average precision decreases dramatically as ranking of the correct item decreases.

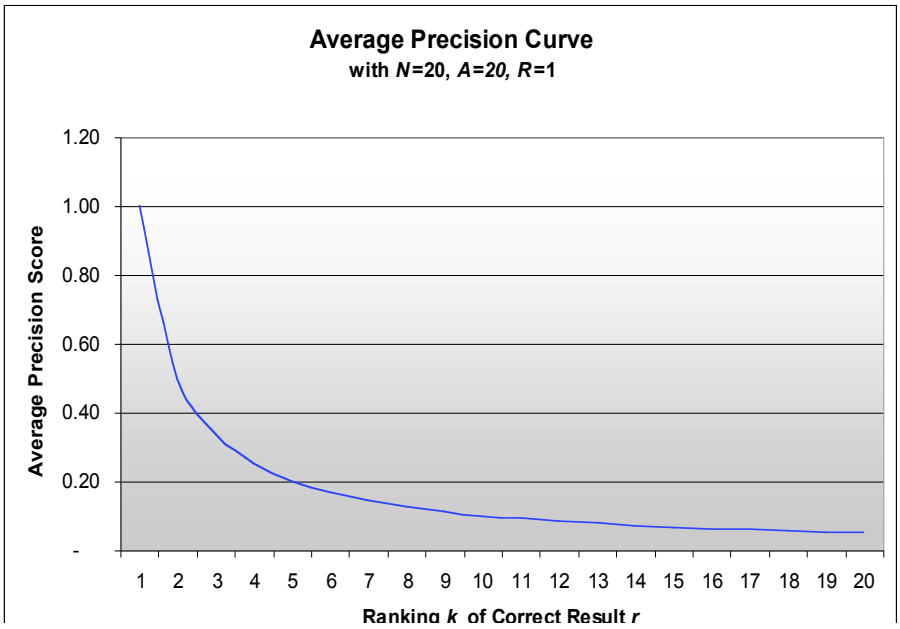


Figure 15. Average precision at different rankings for a set of items with only one relevant result

Average precision is used to evaluate performance for different search strategies for a single request. We use the mean average precision (the average of multiple average precision scores) to evaluate search strategy performance over a set of different requests. In this way we can evaluate whether a new strategy has an overall positive or negative effect when taken over a set of requests.

Significance Testing

It is important to be able to assess how much of the difference in average precision for two searches can be attributed to random variation, and therefore is not significant. We start by calculating the standard deviation, σ_R, of the number of correct results for a result set of size N for a given information request. We calculate σ_R using the formula for variance in a hypergeometric distribution. A hypergeometric distribution is any discrete distribution that describes the number of successes in a series of draws from a finite population without replacement (Triola 2002). When calculating σ_R we also take into account the total number of items in the collection, C, the total number of relevant items, R, and the total number of irrelevant items, I, in the equation

$$\sigma_R = \sqrt{\frac{NRI(C-N)}{C^2(C-1)}} \tag{4.4}$$

Now that we know the standard deviation for the number of relevant results, we can calculate the standard deviation for the precision of the result set, $\sigma_{precision}$, using equation 4.2, giving

$$\sigma_{precision} = \frac{\sigma_R}{N} \tag{4.5}$$

Average precision is the mean of all the precision scores, calculated every time recall increases. For the sake of simplicity, we only calculate the standard deviation of precision when the entire result set is retrieved. In this case $\sigma_{\text{average precision}} = \sigma_{\text{precision}} \div 1 = \sigma_{\text{precision}}$.

Finally, we find the difference between the average precisions of a control and the test search, and calculate the significance of the difference to within 5% by applying a z-test.

4.2 Preliminary Investigation

Prior to our optimisation experiments, we performed an initial investigation into the distinguishing characteristics of the shots that are relevant for the development requests. The analysis of characteristics that distinguish the relevant shots from the irrelevant shots for each request later helped us to develop a number of hypotheses about what type of search strategy would provide the best results for each request.

Specificity and Complexity of the Information Requests

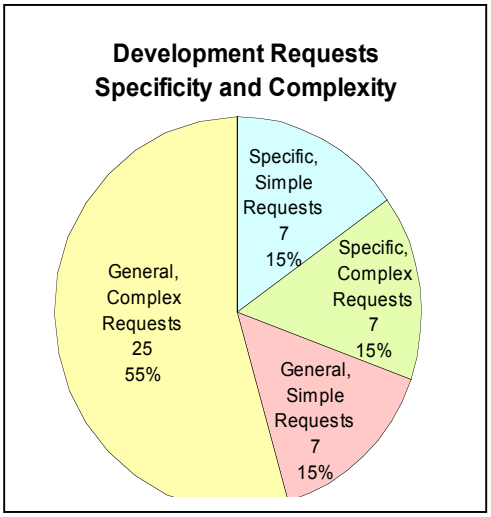


Figure 16. Distribution of request categories in development set

We investigated categorisations of the development requests by AutoSeek using the categorisation criteria from section 3.4.2. Seventy percent of the requests are general, i.e. they do not request an instance of an object, but a class of objects. Seventy percent of the requests are complex, i.e. they refer to more than one named object. Despite the similarity in proportions, there is no strong correlation between the complexity and specificity of a request, as can be seen in Figure 16. Of the 14 specific requests, 7 are complex and 7 are simple. Likewise, of the 14 simple requests, 7 are general and 7 are specific. The bias of the data set is towards general, complex requests, which make up 55% of the requests.

ID	Information Requests Used for Optimisation
0100	Find shots with aerial views containing both one or more buildings and one or more roads.
0101	Find shots of a basket being made - the basketball passes down through the hoop and net.
0102	Find shots from behind the pitcher in a baseball game as he throws a ball that the batter swings at.
0103	Find shots of Yasser Arafat.
0104	Find shots of an airplane taking off.
0105	Find shots of a helicopter in flight or on the ground.
0106	Find shots of the Tomb of the Unknown Soldier at Arlington National Cemetery.
0107	Find shots of a rocket or missile taking off. Simulations are acceptable.
0108	Find shots of the Mercedes logo (star).
0109	Find shots of one or more tanks.
0110	Find shots of a person diving into some water.
0111	Find shots with a locomotive (and attached railroad cars if any) approaching the viewer.
0112	Find shots showing flames.
0113	Find more shots with one or more snow-covered mountain peaks or ridges. Some sky must be visible behind them.
0114	Find shots of Osama Bin Laden.
0115	Find shots of one or more roads with lots of vehicles.
0116	Find shots of the Sphinx.
0117	Find shots of one or more groups of people, a crowd, walking in an urban environment (for example with streets, traffic, and/or buildings).
0120	Find shots of a graphic of Dow Jones Industrial Average showing a rise for one day. The number of points risen that day must be visible.
0121	Find shots of a mug or cup of coffee.
0122	Find shots of one or more cats. At least part of both ears, both eyes, and the mouth must be visible. The body can be in any position.
0123	Find shots of Pope John Paul II.
0124	Find shots of the front of the White House in the daytime with the fountain running.
0125	Find shots of a street scene with multiple pedestrians in motion and multiple vehicles in motion somewhere in the shot.
0126	Find shots of one or more buildings with flood waters around it/them.
0127	Find shots of one or more people and one or more dogs walking together.
0128	Find shots of US Congressman Henry Hyde's face, whole or part, from any angle.
0129	Find shots zooming in on the US Capitol dome.
0130	Find shots of a hockey rink with at least one of the nets fully visible from some point of view.
0131	Find shots of fingers striking the keys on a keyboard which is at least partially visible.
0132	Find shots of people moving a stretcher.
0133	Find shots of Saddam Hussein.
0134	Find shots of Boris Yeltsin.
0135	Find shots of Sam Donaldson's face - whole or part, from any angle, but including both eyes. No other people visible with him.
0136	Find shots of a person hitting a golf ball that then goes into the hole.
0137	Find shots of Benjamin Netanyahu.
0138	Find shots of one or people going up or down some visible steps or stairs.
0139	Find shots of a handheld weapon firing.
0140	Find shots of one or more bicycles rolling along.
0141	Find shots of one or more umbrellas.
0142	Find more shots of a tennis player contacting the ball with his or her tennis racket.
0143	Find shots of one or more wheelchairs. They may be motorized or not.
0144	Find shots of Bill Clinton speaking with at least part of a US flag visible behind him.
0145	Find shots of one or more horses in motion.
0147	Find shots of one or more buildings on fire, with flames and smoke visible.
0148	Find shots of one or more signs or banners carried by people at a march or protest.

Figure 17. The 46 information requests used to develop an optimal search strategy for AutoSeek

Optimal Text Queries

We made use of the relevance feedback method described in section 3.4.5 to analyse 23 requests from the 2003 portion of the development data set. This investigation showed us the ideal textual query terms to use for each information request. The top 5 query terms for each request are shown in Figure 18. These terms were qualitatively investigated to find distinguishing characteristics.

ID	Request Text	Term 1	Term 2	Term 3	Term 4	Term 5
0100	Find shots with aerial views containing both one or more buildings and one or more roads.	city	yesterday	percent	bank	hundred
0101	Find shots of a basket being made - the basketball passes down through the hoop and net.	game	bulls	eighty	extra	jordan
0102	Find shots from behind the pitcher in a baseball game as he throws a ball that the batter swings at.	diamond	run	game	hit	braves
0103	Find shots of Yasser Arafat.	arafat	palestinian	yasser	israel	minister
0104	Find shots of an airplane taking off.	air	airlines	macedonia	albania	airways
0105	Find shots of a helicopter in flight or on the ground.	helicopter	helicopters	highway	nascar	water
0106	Find shots of the Tomb of the Unknown Soldier at Arlington National Cemetery.	tomb	remains	unknowns	vietnam	blassie
0107	Find shots of a rocket or missile taking off. Simulations are acceptable.	space	missiles	chinese	discovery	mir
0108	Find shots of the Mercedes logo (star).	chrysler	merger	mercedes	german	daimler
0109	Find shots of one or more tanks.	tanks	veterans	today	examinations	squandered
0110	Find shots of a person diving into some water.	shot	seemingly	sky	introducing	minutes
0111	Find shots with a locomotive (and attached railroad cars if any) approaching the viewer.	train	trains	vacation	speed	travel
0112	Find shots showing flames.	fires	florida	fire	acres	firefighters
0113	Find more shots with one or more snow-covered mountain peaks or ridges. Some sky must be visible behind them.	climbers	mount	explore	thousands	mountain
0114	Find shots of Osama Bin Laden.	bin	latin	brings	osama	americans
0115	Find shots of one or more roads with lots of vehicles.	rendell	city	cars	traffic	mayor
0116	Find shots of the Sphinx.	egyptian	sphinx	archaeologists	desert	monument
0117	Find shots of one or more groups of people, a crowd, walking in an urban environment (for example with streets, traffic, and/or buildings).	hounded	streets	police	people	students
0120	Find shots of a graphic of Dow Jones Industrial Average showing a rise for one day. The number of points risen that day must be visible.	points	nasdaq	dow	gained	industrials
0121	Find shots of a mug or cup of coffee.	european	data	robust	relies	business
0122	Find shots of one or more cats. At least part of both ears, both eyes, and the mouth must be visible. The body can be in any position.	roasted	chicken	lamb	ocean	select
0123	Find shots of Pope John Paul II.	pope	paul	john	estermann	today
0124	Find shots of the front of the White House in the daytime with the fountain running.	house	white	starr	ammunition	counsel

Figure 18 Top 5 query terms for different information requests

We analysed the feedback terms with respect to the information request, and identified three different types of terms:

- Direct match. Term is contained in the information request.
- Spelling variant. Term is spelling variant of a word in the information request.
- Other. Term is not contained in the information request.

Twelve requests are associated with direct matches in the top 5 feedback terms, as is shown in Figure 19. Six requests are associated with a feedback term that is a spelling variant. For 9 requests, the top 5 terms have no direct relation to the query. Notably, all of the requests that contain proper nouns (specific requests) tend to have at least one of those nouns among the top query terms. For example, the terms “yasser” and “arafat” are both in the top terms for request 0103, “Find shots of Yasser Arafat”. We suggest that this is because nouns describe entities, which are often visual objects. Proper nouns describe specific instances of objects. If an announcer talks about an entity, that entity has an increased likelihood of occurring in a video shot. An announcer is more likely to refer to the specific name of an entity (the proper noun) rather than a more generic term that may include other entities. For example, an announcer is more likely to say “Empire State Building” than “a skyscraper in New York”.

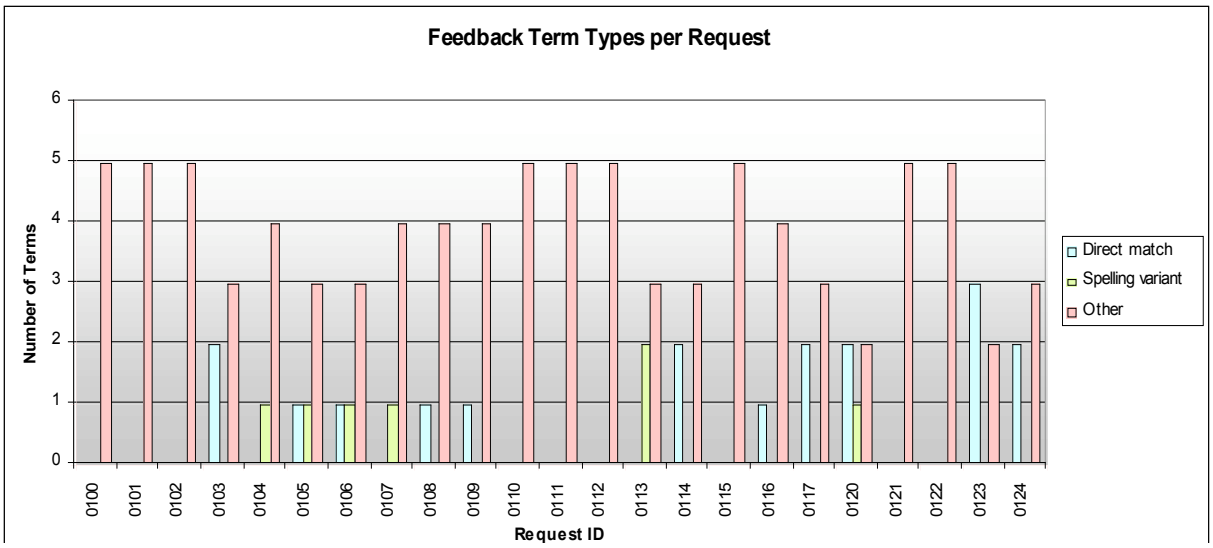


Figure 19. Distribution of the categories of the top 5 feedback terms per request

In examining the feedback terms for requests that were associated with no direct match or spelling variants, we see that they are sometimes related to a particular category, such as the terms “diamond”, “run”, “game” “hit”, and “braves” for request 0102, which asks for shots of a basketball hoop. In other cases, there is no discernable connection between feedback terms and the original request, such as the terms “european", “data, “robust”, “relies”, and “business” for request 0121, which asks for shots of mugs of coffee. Analysis of the relevant shots shows that coffee mugs usually appear in advertisements for business products and are not specifically mentioned in those shots. As commercials do not generally have dialogue that describes the objects in the video shot, this helps to explain the lack of textual correlation.

Optimal Semantic Concept Queries

For the 23 requests from the 2003 developments data we analysed which semantic concept index would provide the best average precision. This was done by retrieving results from each of the available concept indices and calculating the average precision values for each request. The results are shown in Figure 20. For some requests semantic concept index results showed no significant improvement over random sampling of shots from the database. These requests are not shown on the graph.

We can see from the results that for most requests the optimal concept index is intuitively related to the information request. For example, for request 0104 (“Find shots of an airplane taking off”) the best average precision is provided by the “aircraft” index. In a few cases the best semantic concept index does not seem to be at all related to the request, for example for request 0121 (“Find shots of a mug or a cup of coffee”), where the best average precision is provided by the “aircraft” concept. We put this down to coincidence, as the average precision scores here are not very high.

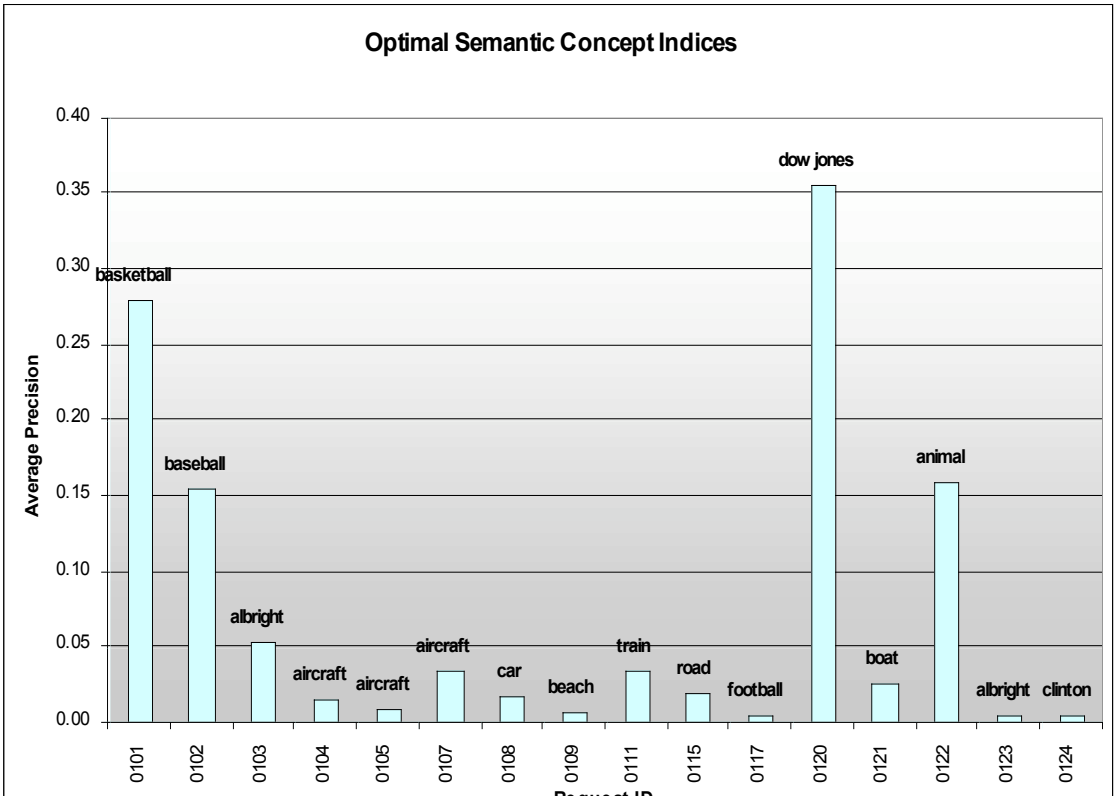


Figure 20. Semantic concept index that provide the best average precision per request

4.3 Experiments

After our preliminary investigation, we performed a number of experiments aimed at achieving the best query formation and result fusion possible within the boundaries of the AutoSeek system. We designed our experiments by using the preliminary investigation results, as well as results from other research, to develop a number of hypotheses about the optimal approach for AutoSeek. We created a test for each hypothesis and assessed the extent to which it held true for different requests in the development set, which enabled us to then provide a number of recommendations for implementation of the AutoSeek system

4.3.1 Query Formation Experiments

We showed in Figure 14 that AutoSeek contains three different types of indices that can be queried: TFIDF, LSI, and semantic concept indices. We experimented to determine the optimal queries, or searches, for each of these indices.

We devised control searches for the TFIDF and the LSI indices, and assessed whether a new query delivered better or worse search results than the control. If the search results showed a significantly higher average precision then the query was deemed better than the control, while if average precision was significantly lower it was deemed to be worse. The control search for the TFIDF index was a TFIDF query on the stopped, stemmed terms of the original request. The control search for the LSI index was an LSI query on the stopped terms of the original request (the LSI index does not incorporate stemming). Each search was limited to a maximum of 1000 results, the maximum number allowed in the TRECVID evaluations. We did not create a control search for the semantic concept indices.

A search on a semantic concept in AutoSeek is very simple: the relevant index is queried and a list of all the shots in the collection is returned, ranked by the pre-calculated probability for each shot that it contains the semantic concept that is desired. We made a decision to focus on choosing only one semantic concept index per search, rather than combining multiple detectors, as previous experience by MediaMill members indicates that the semantic concept indices are not suited for combination. Our experiments in semantic concept query formation focus on deciding which detector to use for which request.

When searching with a TFIDF index, the number of results that are returned can vary. This is problematic for assessment, as this can have a strong effect on average precision. To correct for any distortion due to the number of results, we combined the test search results with the control search results using simple

Borda fusion (weighted Borda fusion with equal weights for all result sets). By doing this, both result sets had the same size. This caused results supported by the test search to be "boosted", changing average precision in a way that could easily be compared to the original control search.

TFIDF Search Queries

We started our investigation of the optimal queries for TFIDF search by investigating the effects of using unstemmed queries in searches. When we stem, different words are reduced to the same root, which means that we will receive more matches than when stemming is not incorporated. However, some of the stemmed matches may be incorrect: for example, the Porter stemming algorithm will reduce both "policy" and "police" to the same root form. Therefore we postulated that an unstemmed match on a term is more likely to be relevant than a stemmed match.

Hypothesis 1. *Boosting results with a TFIDF query to the unstemmed index will improve average precision.*

We tested this hypothesis by performing a search on unstemmed terms. We combined the results with the results of the control search, and evaluated whether there was a significant change in average precision. We found that boosting with unstemmed terms significantly increased average precision in 10 cases, while it significantly decreased average precision in 7 cases. In the remaining 29 cases boosting had no significant effect. We found that the complexity of a request was a good indicator of whether the use of unstemmed results would have a positive effect. When stemming was used for simple queries, average precision either improved or was not significantly affected, and mean average precision increased, as can be seen in Figure 21. When stemming was used for complex queries, the effects were varied, and mean average precision decreased. We suggest that this is because complex requests are more likely to contain plurals and other spelling variations than simple requests. Note that the change in mean average precision appears slight. This is also the case in some other experiments, and is due to the dampening effect of a large number of requests that are unaffected by the new strategy. Due to space considerations we usually show the change in mean average precision rather than individual average precision scores for this and later hypothesis experiments.

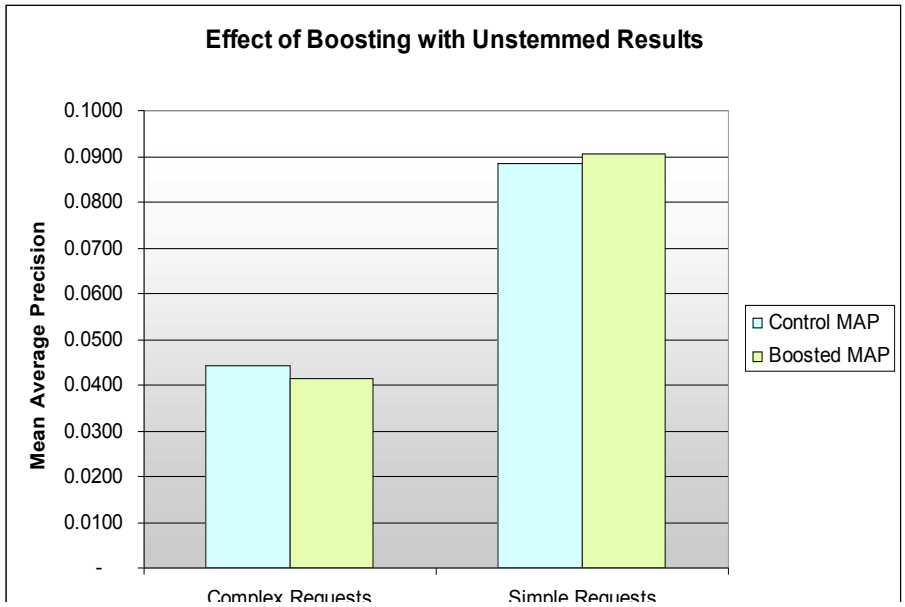


Figure 21. Effect of boosting with unstemmed search results on mean average precision for simple and complex requests

Following our investigation into unstemmed searches, we looked at the effect of focusing on the use of proper nouns in text searches. We saw in our preliminary investigation in section 4.2 that requests that contain proper nouns tend to have at least one of those nouns in the top five query terms. This suggests that queries on these terms will be especially valuable.

Hypothesis 2. *Boosting results with a TFIDF query using only proper nouns will improve average precision.*

To test this hypothesis, we performed a stemmed search using only proper nouns for the 8 requests that contained both proper nouns and other types of words after stopping. The results were combined with the

control search results and analysed to find whether there was a significant improvement in the average precision. We found that searching with common nouns significantly increased average precision in 5 cases, and significantly decreased average precision in 1 case. For the remaining 2 cases there was no significant effect. Mean average precision over the searches for all requests increased, as can be seen in Figure 22. We found that in the one case where boosting had a negative effect, a proper noun had been incorrectly classified as an adjective by the AutoSeek system. Therefore, the hypothesis holds true as long as the part-of-speech tagger correctly classifies all proper nouns.

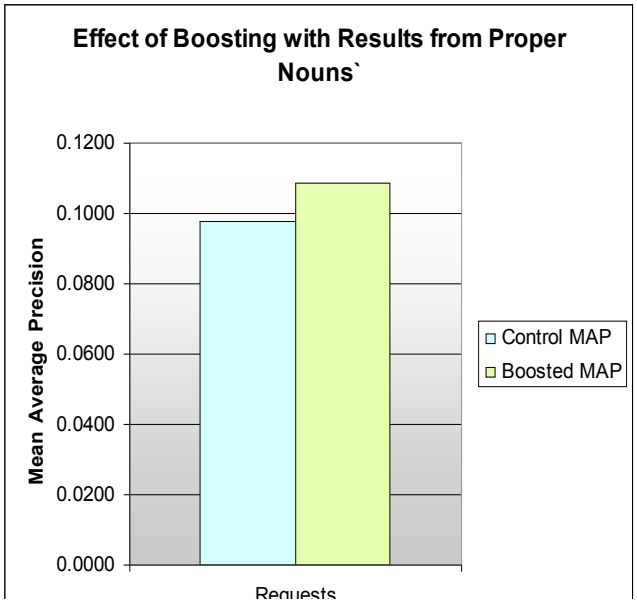


Figure 22. Effect of boosting with results from a search using only proper nouns on mean average precision.

Our preliminary investigation demonstrated that proper nouns are important in text search; it also showed that most of the top search terms tend to be nouns, whether proper or regular. This indicated that we might be able to increase performance by using a search on only nouns.

Hypothesis 3. *Boosting results with a TFIDF query using only the nouns from a request will improve average precision.*

To test this hypothesis, we performed a search using only nouns for the 35 requests that contained both nouns and other types of words after stopping. We found that boosting significantly increased average precision in 7 cases, and it significantly decreased average precision in 1 case. In the remaining 29 cases boosting had no significant effect. In the one case where noun search had a negative affect on average precision a noun had been incorrectly classified as an adjective by the AutoSeek system (the same noun that was incorrectly classified in Hypothesis 2). This indicates that while searching on nouns does not usually have a significant effect, it does sometimes increase performance when part-of-speech tagging has been performed correctly.

LSI Search Queries

By the same logic used in Hypothesis 2, we expect that proper nouns will produce the best search results for an LSI search.

Hypothesis 4. *The use of an LSI query on only proper nouns will increase average precision.*

For each of the 8 requests containing both proper nouns and other words after stopping, we performed a search using only the proper nouns as identified by part-of-speech analysis. We created a third result set by fusing the control and test search. We found that the results were comparable to the results for a proper noun search with TFIDF in that searching on only proper nouns usually had a positive effect. The new search significantly increased average precision in 6 cases and significantly decreased average precision in 1 case, while in the remaining case there was no significant change. Furthermore, as can be seen in Figure 23, average precision increased further when this search was boosted with the control search.

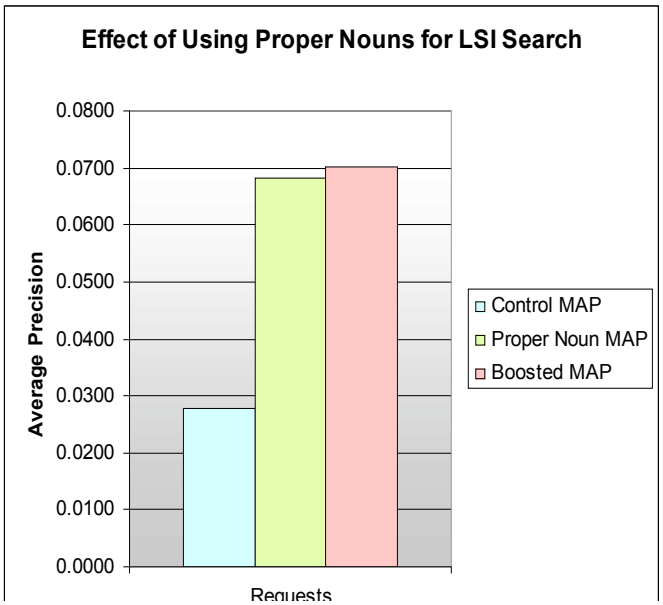


Figure 23. Effects of searching on proper nouns only, and of combining those search results with a regular LSI search

Besides investigating the effect of proper nouns on LSI search, we also investigated the effect of creating a query using all nouns on LSI search. Once again, the reasoning was the same as that used in developing one of the TFIDF hypotheses, Hypothesis 3. Nouns seem to occur more frequently among optimal query terms than other types of words.

Hypothesis 5. *The use of an LSI query on only nouns will increase average precision.*

The test search was performed using all of the nouns from the original request text. There was a slight increase in performance when searching on only nouns, while average precision increased significantly in 4 cases and decreased significantly in 1 case. For the remaining 32 cases using only nouns to search had no significant effect. We did not see any improvement in average precision from combining control search results with noun search results. We saw that for requests with proper nouns, the search results from Hypothesis 4 were better than the results we saw here.

The last hypothesis that we investigated for LSI search was not related to TFDIF search. As indicated in section 3.3.1, LSI operates by first creating a multi-dimensional space where each different word is one dimension, and subsequently using principal component analysis to reduce the number of dimensions. The number of dimensions finally created can be as low as one, at which point all information differentiating different words is completely lost. Alternatively, the number can be as high as the number of unique words in the entire collection. At this point, all information about the differences between words is completely retained. We postulate that the more information we retain about the differences between words, the better results will be.

Hypothesis 6. *The use of a high number of dimensions will increase average precision.*

There were over 18,000 unique terms in the development collection, which meant that potentially we could create an LSI index with over 18,000 dimensions. Unfortunately, due to memory and processing restrictions, our LSI module allowed a maximum of only 1,000 dimensions and a minimum of 300 dimensions, so we were restricted to experimenting within this range. To test the hypothesis we performed LSI searches on the stopped request test, increasing the dimensions in increments of 100, and comparing the effects of the change of dimensionality per request. We found that average precision generally increased as the number of dimensions used increased as can be seen in Figure 24, which shows the average precisions of individual requests in black, and the overall mean average precision in red. We can see that the average precision tends to increase as the number of dimensions used increases. There are some notable exceptions, such as the line that peaks at 600 dimensions. This line is for topic 0147. Analysis did not reveal why some requests peak at a lower dimensionality, but overall mean average precision increases as dimensionality increases. This indicates that within the narrow range of dimensions available within the AutoSeek system the highest possible number of dimensions is usually preferable.

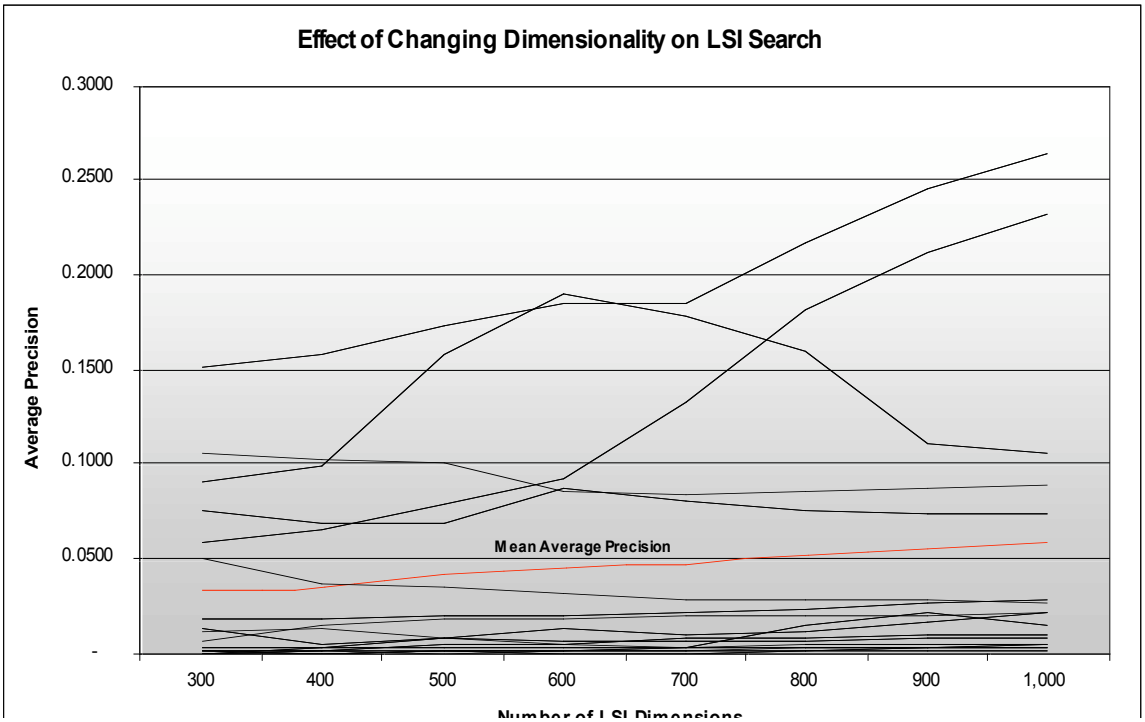


Figure 24. Effect of changing number of LSI dimensions on requests, showing Mean Average Precision over all requests

Semantic Concept Search Queries

As we mentioned earlier in this section, our optimisation experiments for semantic concept search focus on finding the single best concept index to use in the search. We use WordNet to link semantic concepts from the text to semantic concept indices. To accomplish this we retrieve the entire hypernym and hyponym relationship trees for each semantic concept in the text, and check whether they contain a semantic concept that is connected with a semantic concept index. We did not include the meronym and holonym relationship trees as investigations very early on in our research had indicated that these relationships quickly expand past the initially detected concept. For example, a meronym of the concept "aircraft" is "skeletal frame", which is also a meronym of the concept "building". Another factor in our choice not to use the meronym and hyponym relationships is the inconsistency with which they are annotated in WordNet. For instance, "aircraft" has over 20 meronyms listed in WordNet, while "helicopter" only has 4 ("skeletal frame" is not one of them).

In section 3.4.2 we outlined the problem of ambiguous WordNet nouns: words that are extracted from the text but have more than one meaning. A considerable number of WordNet nouns have multiple meanings, and our first experiment was simply to determine the best method for disambiguating WordNet nouns. We have previously discussed specialised disambiguation methods employing similarity measures especially developed for WordNet, and we also discussed a very simple disambiguation method by which the most frequent meaning of a word is simply selected. We investigated this to see which method would work best.

Hypothesis 7. *A specialised disambiguation method will retrieve better semantic concept indices than a frequency-based disambiguation method*

To test this hypothesis we qualitatively determined the most relevant semantic concept index or indices for each request. For example, for the request for "shots of a helicopter in flight or on the ground" we selected the 'aircraft' detector as being most relevant for the request. We then disambiguated the words of the requests using the different disambiguation strategies described in section 3.4.2, including the frequency-based method. Next we identified all semantic concept indices that were connected to the request in the hyponym and hypernym trees that were associated with each WordNet concept in the request. Finally, we checked whether the most relevant semantic concept index was found in the hypernym or the holonym tree.

We identified a total of 20 requests that were related to one or more semantic concept indices, and in total there were 24 semantic concept indices that we qualitatively determined to be the most relevant to the requests. Our results are shown in Figure 25. We found that the frequency-based method, shown as "default" , demonstrated the best performance, retrieving the most relevant indices for 24 out of 25 requests. This surprising result is possibly due to the disambiguation methods not being designed for the small number of

concepts that could be extracted from the queries. It is also possible that the formulations of the evaluation requests are expressly kept simple by the TRECVID committee, and that for this reason simple disambiguation works best.

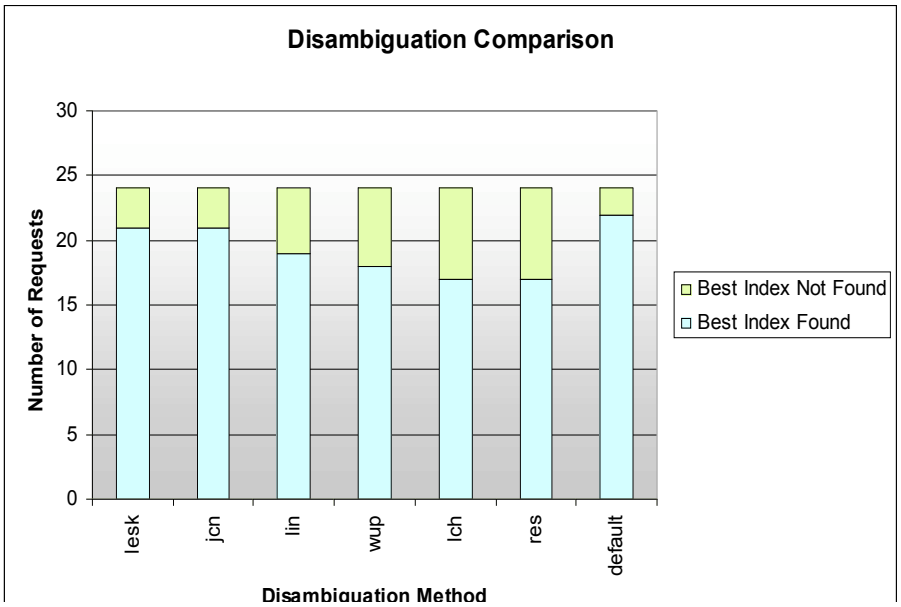


Figure 25. Comparison of the different disambiguation strategies contained in AutoSeek

The hypernym and hyponym trees retrieved for the concepts in a request often also match to other concept indices than the most relevant one. Consider, for example, a request for "basketball". This will match the "basketball" index, but also the "sport" index. Intuitively, the semantic concept index that is the most similar to one of the terms in the information request will be the most relevant.

Hypothesis 8. *Searching on the semantic concept index that is the most similar to the terms in the request will produce the best average precision.*

We used the Resnik similarity measure to measure similarity between each related concept index that we found, using the simple disambiguation strategy that was shown to perform best in the previous hypothesis section. We then assessed whether that concept index delivered better average precision than results from other indices. In this case our expectations were confirmed. For 16 requests the most similar concept index provided the best average precision. For 2 requests there were multiple most similar indices, one of which provided the highest average precision. In 1 case the most similar semantic concept did not provide the highest average precision, and in 1 case all of the semantic concept indices produced an average precision of 0.

4.3.2 Query Formation Strategy

The final query formation strategy developed through our experiments is summed up in Figure 26.

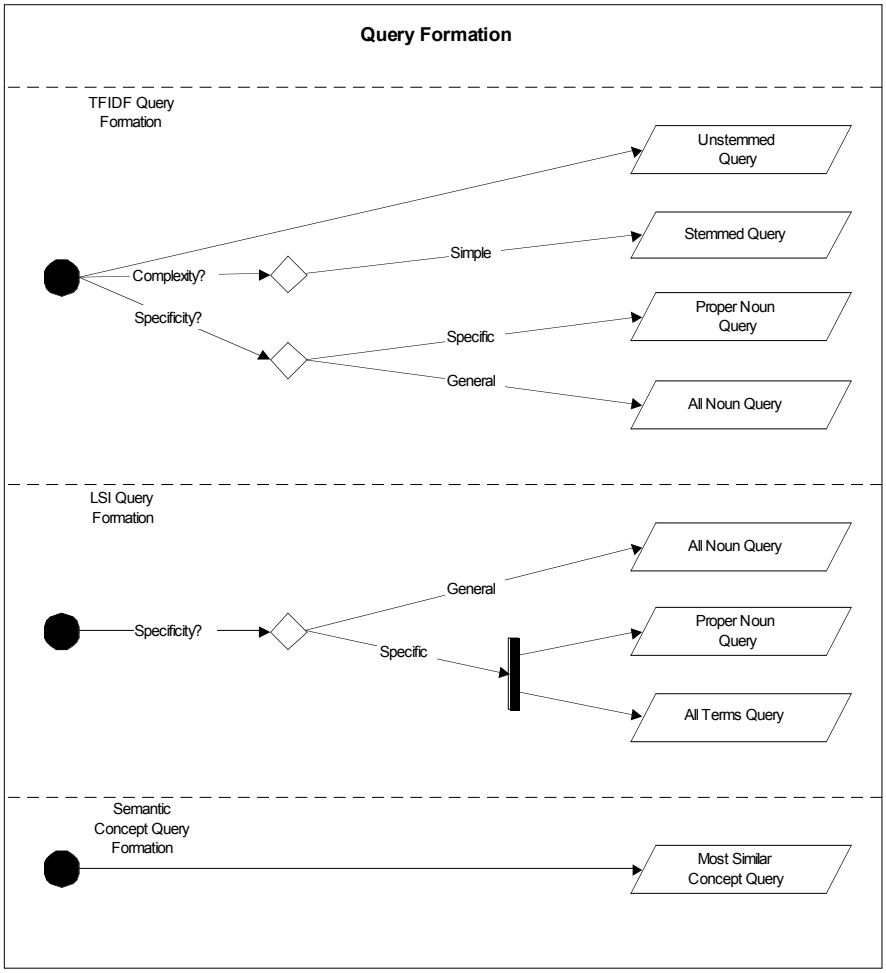


Figure 26. Query formation strategy

In TFIDF query formation we start with a stemmed query on the request terms with stop words removed. The results from Hypothesis 1 indicate that we should create an additional query using the unstemmed request terms if the request is simple. If the request is specific (i.e. contains proper nouns), then we form a query containing only the proper nouns and add it to the results, as shown by the results from Hypothesis 2. Otherwise, the results from Hypothesis 3 motivate us to create a query using only nouns.

In LSI query formation, we are motivated by the results from Hypothesis 4 to form a query from all the request terms with stop words removed if the request is specific. In this case we also form a query using only the proper nouns. For general requests we create a query with all of the nouns in the request without any other word types, as indicated by the experimentation for Hypothesis 5. All searches should be carried out at the maximum available number of dimensions, 1000, as results from Hypothesis 6 show us that this will give the highest mean average precision

When choosing our most similar concept query we use a simple frequency-based disambiguation strategy for WordNet nouns in the request, as the results from Hypothesis 7 indicate that this will find the most results. We then search for related concept indices have been found using the hypernym/hyponym relationship in WordNet and select the most similar index using the Resnik similarity measure to be queried, as this has been shown in the experiments for Hypothesis 8 to be a good indicator of index relevance. It is possible that more than one index has the highest similarity, we did not investigate this further, but in this case we simply fuse all of the most similar indexes using Borda fusion with equal weights.

Evaluation

We performed an evaluation of the final search strategies for TFIDF and LSI search and found that for TFIDF search the combination of recommendations significantly increased average precision in 10 cases, and

significantly decreased average precision in 1 case. In the remaining 35 cases there was no significant effect. Mean average precision over all results increased from 0.0577 to 0.0613 over all of the cases, as can be seen in Figure 27. The requests for which combination of recommendations had a significant effect are shown in Figure 28. Here we can see that the one request for which the combination of strategies had a negative effect, with request id 0106, shows a decrease in average precision of 0.5. This decrease of average precision is due to the incorrect identification of a noun as an adjective by the AutoSeek system.

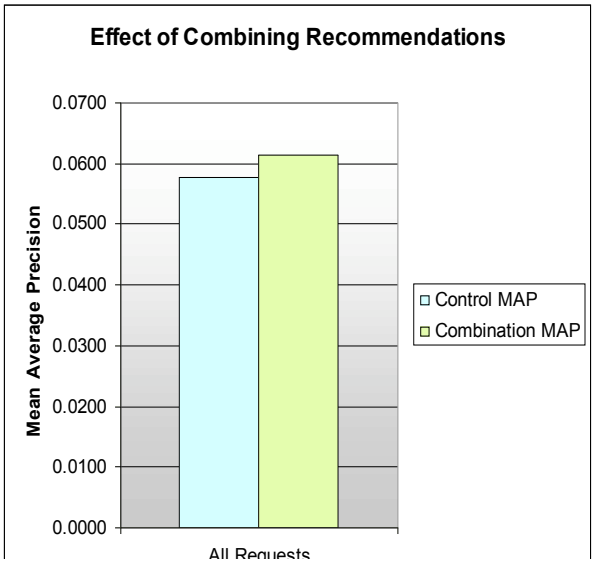


Figure 27. Effect of the final TFIDF query formation strategy on mean average precision over all requests

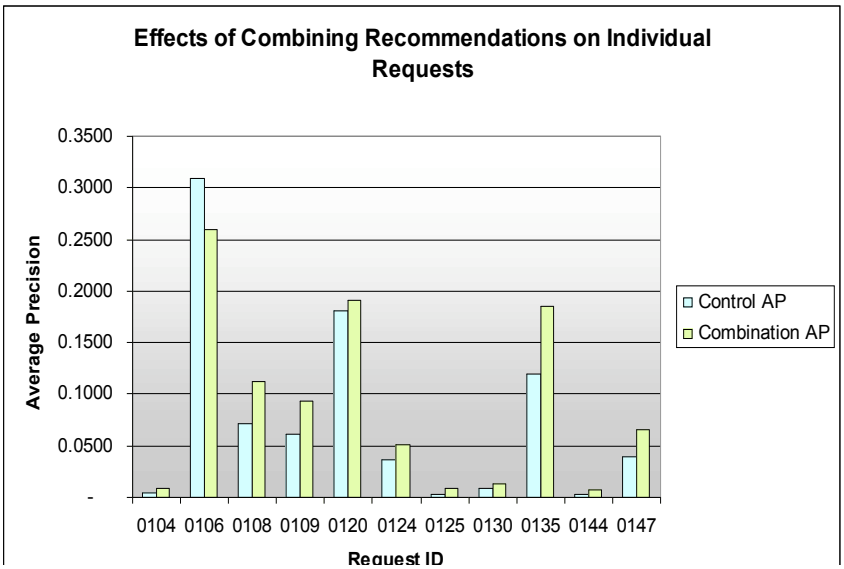


Figure 28. Effect of the final TFIDF query formation strategy on individual requests (only significantly affected requests shown)

For LSI search we found that the combination of strategies significantly increased average precision in 27 cases and significantly decreased average precision in 2 cases. In the remaining 17 cases combination had no significant effect. Mean average precision over all results increased from 0.0302 to 0.0777 over all of the cases, as can be seen in Figure 27. These results indicate that the implementation of the recommendations for TFIDF and LSI search have a considerable positive effect on engine performance as measured by average precision.

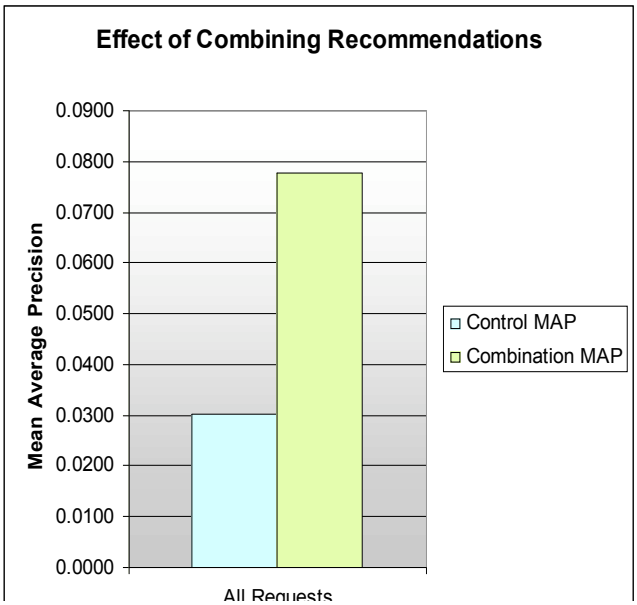


Figure 29. Effect of the final LSI query formation strategy on mean average precision over all requests

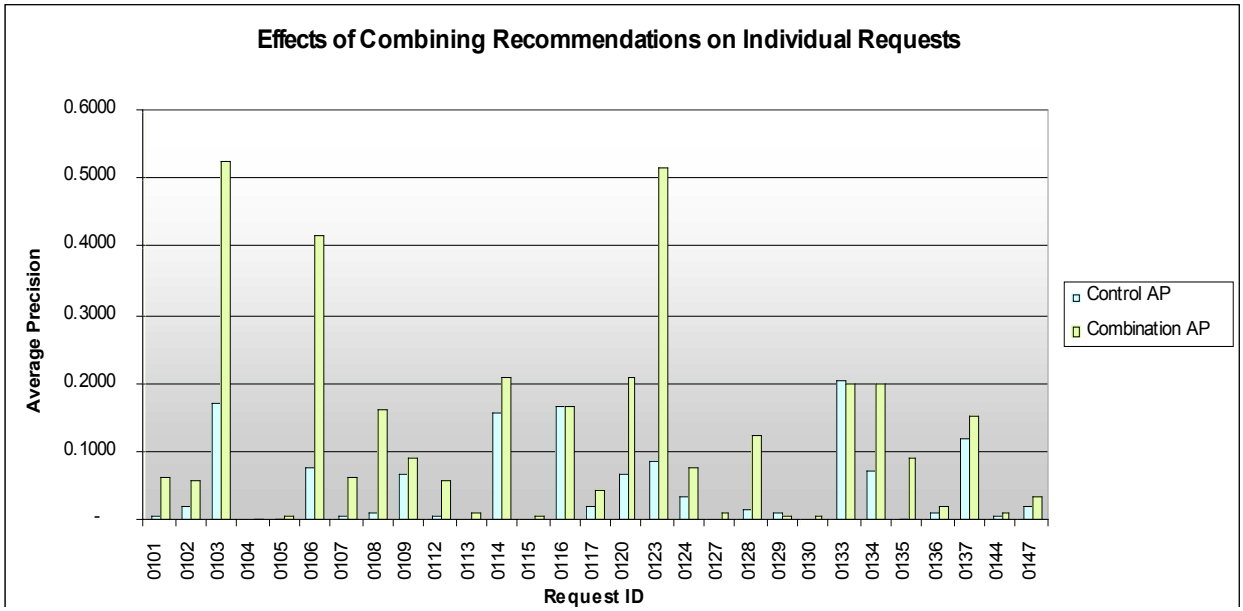


Figure 30. Effect of the final LSI query formation strategy on individual requests (only significantly affected requests shown)

4.3.3 Result Fusion Experiments

Our result fusion optimisation experiments focus on the combination of result sets from the three different types of indices: TFIDF, LSI, and semantic concept indices. We perform an initial fusion step to combine results from multiple queries for one type of index, where all result sets are given an equal weighting, as outlined in our evaluation of the query formation strategy.

We first focussed on the combination of results from the text indices to provide a final set of text results. We posited that because TFIDF search results are directly related to the words in the request they will suffer from less 'noise', and are more likely to be correct than LSI results. By giving TFIDF results a high weighting, we would give preference to correct results.

Hypothesis 9. *Giving TFIDF search the highest weighting possible is the best way to fuse TFIDF and LSI search results.*

We tested this hypothesis by using weighted Borda fusion to combine the results from the TFIDF indices and the LSI index across a range of weighting pairs. We selected the values for the weighting pairs such that the sum was always 1, and calculated the average precision at weight increments of 0.02. We determined average precision for each weighting combination and plotted this information on a graph. Relationships such as specificity and complexity were investigated find out if there were any differences in the types of weightings that should be used.

We found that for 22 requests the optimal weighting scheme had a higher TFIDF weighting than LSI weighting. For 12 requests the optimal weighting pair had a higher LSI weighting than TFIDF weighting. In 12 cases average precision was not significantly influenced by the weighting used. Requests were examined on an individual basis, and a relationship was found between the complexity and specificity of the request and the best type of fusion to use. An overview of this relationship can be seen in Figure 31, where the mean average precision for different categories of information requests is plotted against different weightings. We see here that complex requests, both specific and general, tend to peak at a TFIDF weighting of about 1, and an LSI weighting of 0. Simple requests benefit most from an approximately equal weighting of TFIDF and LSI results. Overall, the optimal weighting pairs for different categories of requests are:

- General, complex requests: TFIDF = 1.00 LSI = 0.00
- Specific, complex requests: TFIDF = 0.94 LSI = 0.06
- General, simple requests: TFIDF = 0.48 LSI = 0.52
- Specific, simple requests: TFIDF = 0.56 LSI = 0.44

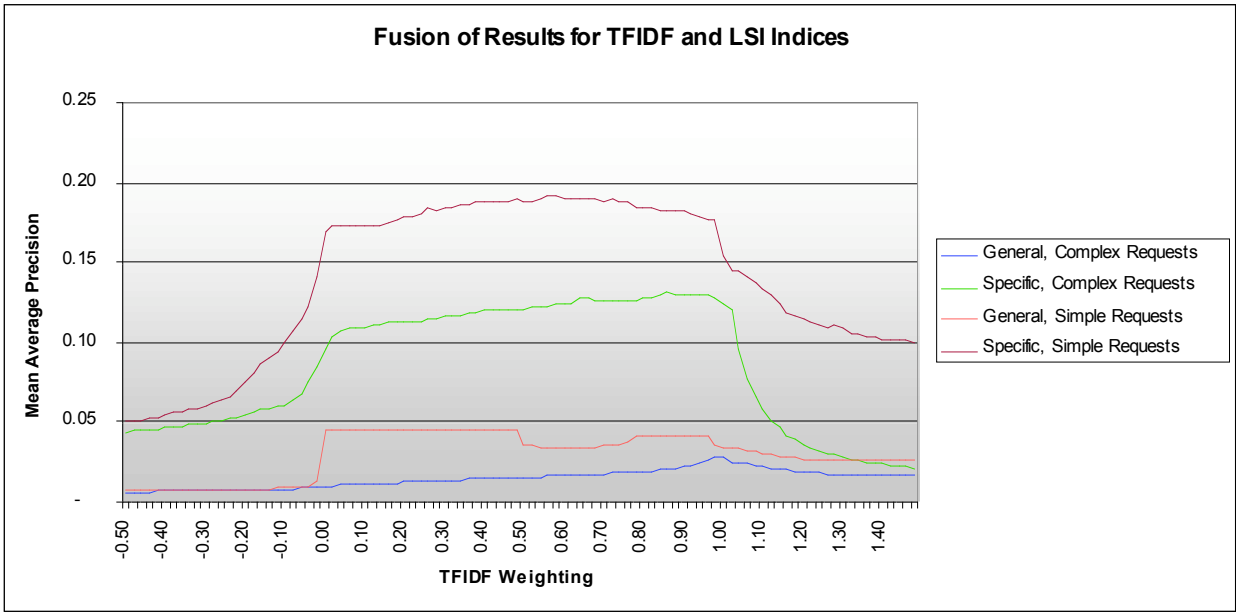


Figure 31. Weighted Borda Fusion of optimal TFIDF and LSI search results over different weights

Next we investigated the effect of using text as a preliminary filter for concept search. We were guided by the findings of Informedia outlined in section 2.4. Their research indicated that text is the best indicator of the relevance of a shot. According to these results, concept search should be considered secondary to the text search.

Hypothesis 10. *Using concept search to re-rank results of a text search will provide better results than using a concept search over all shots in the collection.*

To test this hypothesis, we took two different approaches for concept search. For the control search, a search was carried out on the best matching concept. If more than one concept had the highest similarity score, the searches from each were combined with equal weightings. For the test 'filtered' search, the results from the text search were re-ranked using their concept scores. We tested for significant differences between average precisions of the filtered search and the control search, and found that re-ranking text results with concept search provided better average precision than a pure concept search in 8 cases, while it provided worse average precision in 8 cases. It had no significant effect in 5 cases. Analysis on an individual request basis showed that when the semantic concept was directly matched in the text, filtering did not improve average precision. When

the semantic concept was quite far removed from the text, and the text was quite specific, filtering had a positive effect. We therefore used Resnik's measure of information content to analyse the information content of the individual concepts in the information request and the similarity of the best semantic concept index to the request. We found that filtering only resulted in an improvement when the similarity of the concept to the request was low (a Resnik similarity of less than 8), and the request contained a concept with a relatively high information content (a Resnik information content of more than 8).

Subsequently we explored the best fusion strategy for merging the text results and the concept results. Experiments from past TREC evaluations have shown that text is the best indicator for search. Therefore, by weighting text more highly than concepts average precision should improve.

Hypothesis 11. *Giving text a higher weighting than concepts will result in better average precision.*

To test this hypothesis the merged text results and the concept results were combined using weighted Borda fusion with a number of weighting pairs, across the same range and at the same increments as those used in testing Hypothesis 9. Average precision was determined for each weighting pair and different relationships such were analysed to find out if there were any patterns in the types of weightings produced optimal results. We identified an experimental set of relationships between the semantic concept indices and the semantic concepts in text, using the Resnik measures of concept similarity and information content outlined in section 3.2.2: In this set of relationships we also took into account whether an index was an exact match to one of the request concepts, or if it was retrieved from a hypernym or hyponym tree.

1. **Closely Related and Highly Informative Index.** The index is very similar to a concept in the information request and has a high information content value. This entails that the concept is very relevant to the request, and also quite rare, and so should be given a very high weighting for fusion.
2. **Closely Related or Highly Informative Index.** The index is either very similar to a concept in the information request or has a high information content, but not both. This entails that the concept is likely to be of value to the request, but not extremely so.
3. **Low Value Index with a Direct Concept Match.** The index directly matches a concept in the query, but due to a low level of specificity it is not considered very similar to or informative for the request.
4. **Low Value Index without a Direct Concept Match.** The index has been found in a hypernym or hyponym tree, but is of little relation to the original request concepts.

We developed an ad hoc logic for categorisations of different kinds of relationships by analysing the similarity and information content scores of the semantic concept indices, as well as results of the fusion. The logic is shown in Figure 32.

Related **and** Informative:	Concept Information Similarity > 12.5 AND Concept Information Content > 12.5
Related **or** Informative:	Concept Information Content > 8 OR Concept Information Content – Request Information Content < 2
Low Value **and** Direct Match	Concept is direct match AND Concept Information Content < 8 AND Concept Information Content – Request Information Content > 2
Low Value **no** Direct Match:	Concept is not direct match AND Concept Information Content < 8 AND Concept Information Content – Request Information Content > 2

Figure 32 Logic used in categorising concepts

In 14 cases the optimal weighting scheme had a higher concept weighting than text weighting. In 7 cases the optimal weighting schemes had a higher text weighting than concept weighting. The result for fusion of different categories of requests is shown in Figure 33. We see that relevant and informative requests indexes prefer a very high concept weighting, while relevant or informative indexes benefit from a combination of indices. Low value indexes did not improve average precision. When using the categorisation outlined in the approach, the best weighting was:

- Closely Related and Highly Informative Index: Concept = 1.00 Text = 0.00
- Closely Related or Highly Informative Index: Concept = 0.82 Text = 0.18
- Low Value Index and Direct Match: Concept = -0.02 Text = 1.02
- Low Value Index no Direct Match: Concept = 0.00 Text = 1.00

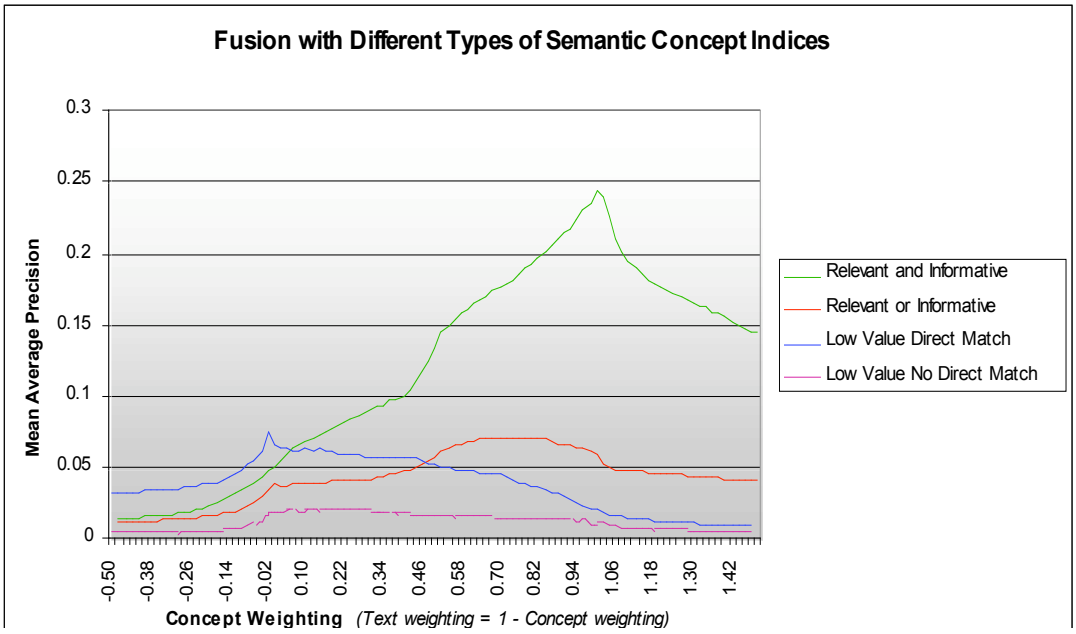


Figure 33. Fusion of different types of semantic concept indices using the criteria developed in the approach

Finally, we investigated the effect of filtering shots containing an extremely common concept, the news anchor, from the search results. People generally do not search for news anchors, yet a large portion of the shots in the data set contain them. By using the "anchor" semantic index to remove shots that are almost certain to contain anchors, we should be able to remove incorrect shots and increase average precision.

Hypothesis 12. *Removing anchor shots from results will improve performance for all types of search.*

To test this hypothesis we performed a control search using the optimal strategy defined by previous hypotheses. We then created a new set of results by removing shots classified as containing an anchor from the control search. Anchor shots were defined as those shots that were given a probability of containing an anchor of greater than 0.40 by the anchor semantic concept index. We found that removal of anchor shots increased average precision in 13 cases, and decreased average precision in 6 cases. It had no significant effect in 27 cases. Qualitative analysis showed that in the cases where removing anchors decreased average precision, this was often due to incorrect shot segmentation, which caused an anchor and a news story to be included in a single shot.

4.3.4 Result Fusion Strategy

The final query formation strategy developed through our experiments is summed up in Figure 34.

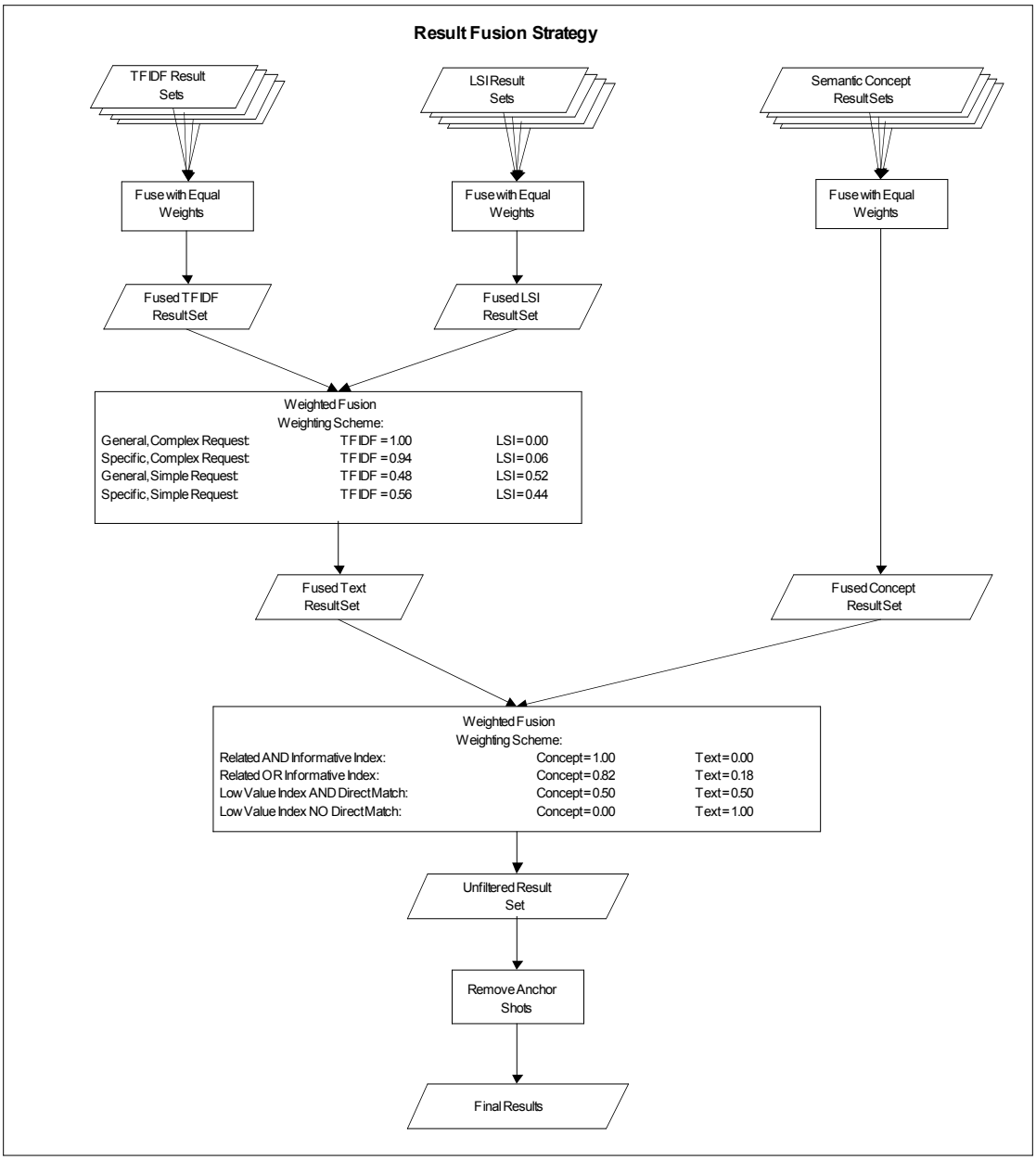


Figure 34. Result fusion strategy

Our experiments demonstrated that specificity and complexity play an important role for fusing results from TFIDF and LSI indices. The optimal weights for different categories of requests are outlined in Hypothesis 9. When performing concept search, we employ a pure concept search than using the concept index to re-rank text results, unless the most similar concept is not very similar to the topic (Resnik similarity of less than 8), and the most similar concept is not very informative (Resnik information content of less than 8). This is shown in the experiments for Hypothesis 10. In the results for Hypothesis 11 we outlined the optimal weighting scheme for text and semantic concept search results according to information content of the different concepts, and the similarity between concepts from the request and the related semantic concept indices[17]. Finally, experiments for

[17] In the final search strategy shown here, the weighting for indices with a low value but direct concept match has been changed from -0.02 to 0.50. We did this because we expected that due to the addition of foreign language channels, the text in the test set would be less valuable than the text in the development set. This is the only manual change that we made.

Hypothesis 12 led us remove anchor shots from the final result set, especially for data sets where when anchor detection and shot segmentation are of high quality.

4.4 Final Experiment Evaluation

To confirm the validity of the fusion recommendations, we performed an evaluation of the final recommendation strategy to see if the new search strategies outperformed a control search. As the control search we use the top 1000 results from an LSI search on all of the words contained in the information request.

The final strategy significantly increased average precision over the text baseline in 37 cases, as can be seen in Figure 35. It decreased average precision in 1 case. Average precision was not significantly affected in 8 cases. Mean average precision increased from 0.0442 to 0.1044, as can be seen in Figure 36.

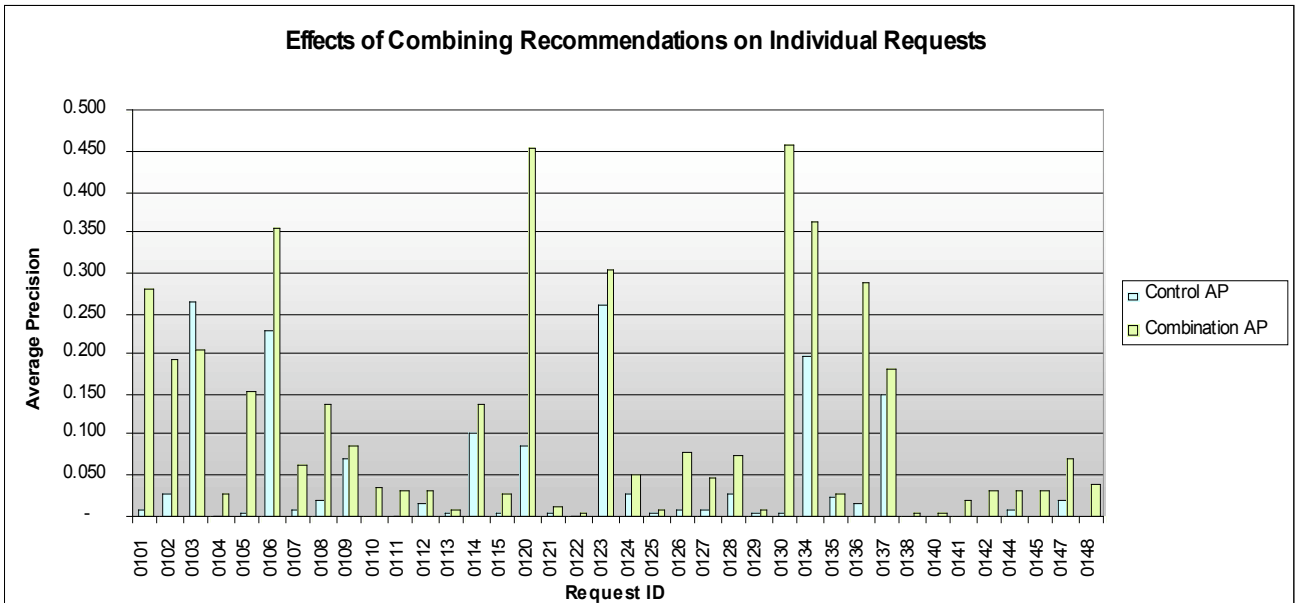


Figure 35. Average precision of baseline and final combination of recommendations per request (only significantly affected requests are shown)

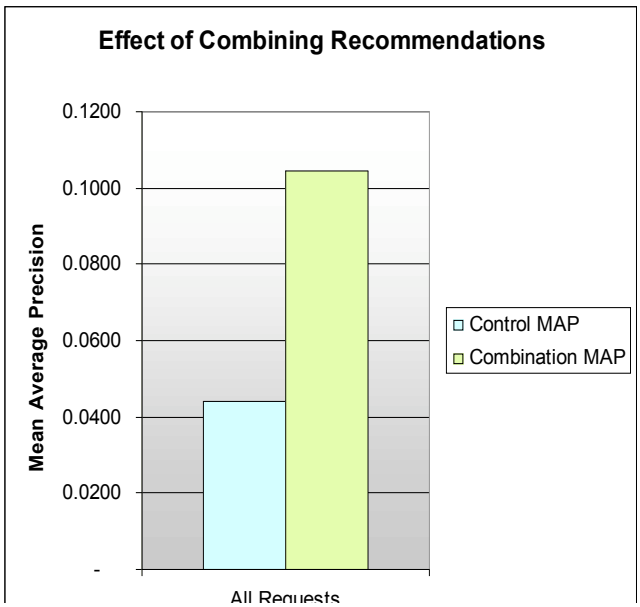


Figure 36. Mean average precision of control and combination searches over all requests

CHAPTER 5 EVALUATION

In September of 2005 we submitted AutoSeek to external assessment in the TRECVID 2005 automatic search task[18]. In this task each participant was given 24 multimodal statements of information need whereby each statement consisted of a text request and a number of multimodal examples; the text requests are shown in Figure 37. Each participant then performed automatic searches for the statements and submitted the results to the TRECVID committee, who pooled the top ranked results from the different searches and manually checked every shot for relevance to the information that was required. In this way they created a ground truth, which was then used to calculate the average precision of every set of search results. This allows an objective comparison between different participants in the automatic search task. To the best of our knowledge, we were the only automatic search entrant that did not make any use of the multimodal examples that were provided (this can not be confirmed until the final versions of the TRECVID papers become available on the TRECVID website in March 2006).

ID	Information Requests Used for Optimisation
0149	Find shots of Condoleeza Rice.
0150	Find shots of Iyad Allawi, the former prime minister of Iraq.
0151	Find shots of Omar Karami, the former prime minister of Lebannon.
0152	Find shots of Hu Jintao, president of the People's Republic of China.
0153	Find shots of Tony Blair.
0154	Find shots of Mahmoud Abbas, also known as Abu Mazen, prime minister of the Palestinian Authority.
0155	Find shots of a graphic map of Iraq, location of Bagdhad marked - not a weather map.
0156	Find shots of tennis players on the court - both players visible at same time.
0157	Find shots of people shaking hands.
0158	Find shots of a helicopter in flight.
0159	Find shots of George W. Bush entering or leaving a vehicle (e.g., car, van, airplane, helicopter, etc) (he and vehicle both visible at the same time).
0160	Find shots of something (e.g., vehicle, aircraft, building, etc) on fire with flames and smoke visible.
0161	Find shots of people with banners or signs.
0162	Find shots of one or more people entering or leaving a building.
0163	Find shots of a meeting with a large table and more than two people.
0164	Find shots of a ship or boat.
0165	Find shots of basketball players on the court.
0166	Find shots of one or more palm trees.
0167	Find shots of an airplane taking off.
0168	Find shots of a road with one or more cars.
0169	Find shots of one or more tanks or other military vehicles.
0170	Find shots of a tall building (with more than 5 floors above the ground).
0171	Find shots of a goal being made in a soccer match.
0172	Find shots of an office setting, i.e., one or more desks/tables and one or more computers and one or more people.

Figure 37. Textual information requests in the TRECVID 2005 automatic search task

5.1 Test Data

The data set used in the TRECVID 2005 evaluation was similar to the data set described in 3.1 in that it concerned a large body of captured news video, but was significantly different in a number of ways:

Video Sources: The development set contained video fragments from the CNN Headline News and ABC World News Tonight channels. The test set contained video fragments from the Arabic LBC, the Chinese CCTV4 and NTDTV, and the American CNN and NBC channel.

Language: The development set contained only English-language video. The test set contained English, Arabic, and Chinese language video. The ASR transcript of each video was provided by a commercial off-the-shelf speech recognition program, and the ASR transcript of non-English videos was then translated to English using machine translation (also provided by an off-the-shelf product).

Time Frame: The development set contained video recorded in 1998. The video data for the test set was recorded in November of 2004.

Concept Indices: The development set had 28 semantic concept indices associated with it. The test set contained 88 different semantic concept indices.

We performed an analysis of the 24 assessment information requests shown in Figure 37. Like the development information requests, they represent a mixture of requests for people, objects, and situations.

[18] http://www-nlpir.nist.gov/projects/tv2005/tv2005.html

The distribution of the different categories of the requests is similar to the distribution for the development set, as can be seen in Figure 38, which shows the AutoSeek categorisations of requests in the development and the test sets. The majority of requests in both sets are general and complex, and the remaining requests are divided between the remaining category combination possibilities.

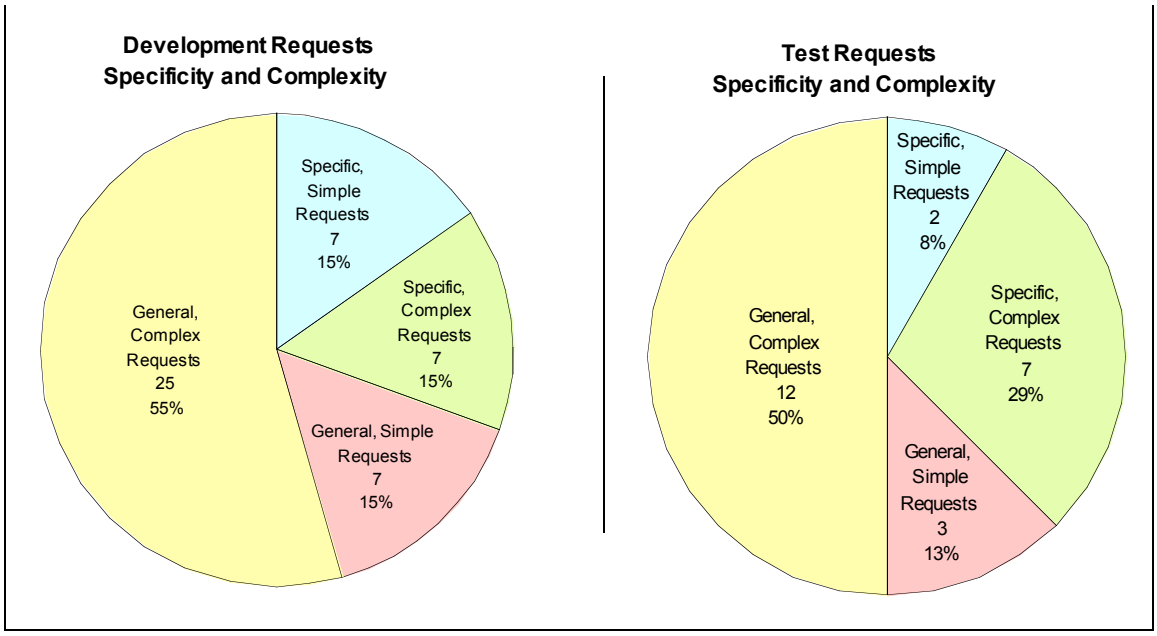


Figure 38. Category distributions for requests in the development and test sets

5.2 Results

We entered two sets of search results, or runs, in the TRECVID evaluation. One of these runs was a baseline search that used only an LSI query on the stopped text of each information request to form results, and the other was an automatic search using the full optimised AutoSeek search strategy for each request. The results of the evaluation are shown in Figure 39. Our system performed very well for a number of topics. We gained the highest average precision for request 0163, and performed better than the median for 13 out of the 24 requests. We show the ranking of the AutoSeek system compared to the other 10 teams on a per-request basis in Figure 40, where is apparent that there is a large variation in the ranking of the AutoSeek system. We gain the highest score for one request, rank second for 3 requests, rank third for 2 requests. On the other hand, we gain among the lowest three ranks for 7 requests. Overall, when looking at mean average precision (MAP) over all requests, AutoSeek ranks fourth over all systems.

To gain insight into the performance of the AutoSeek system in the TRECVID evaluation, we performed two unofficial evaluations using the ground truth information that became available afterwards. The two additional evaluations were of the final optimised strategy using only text search, and the final optimised strategy search using only concept search. These are contrasted against the baseline and the final AutoSeek search strategy (which integrates text and semantic concepts) in Figure 41. We will analyse the performance of the different types of search in the following sections.

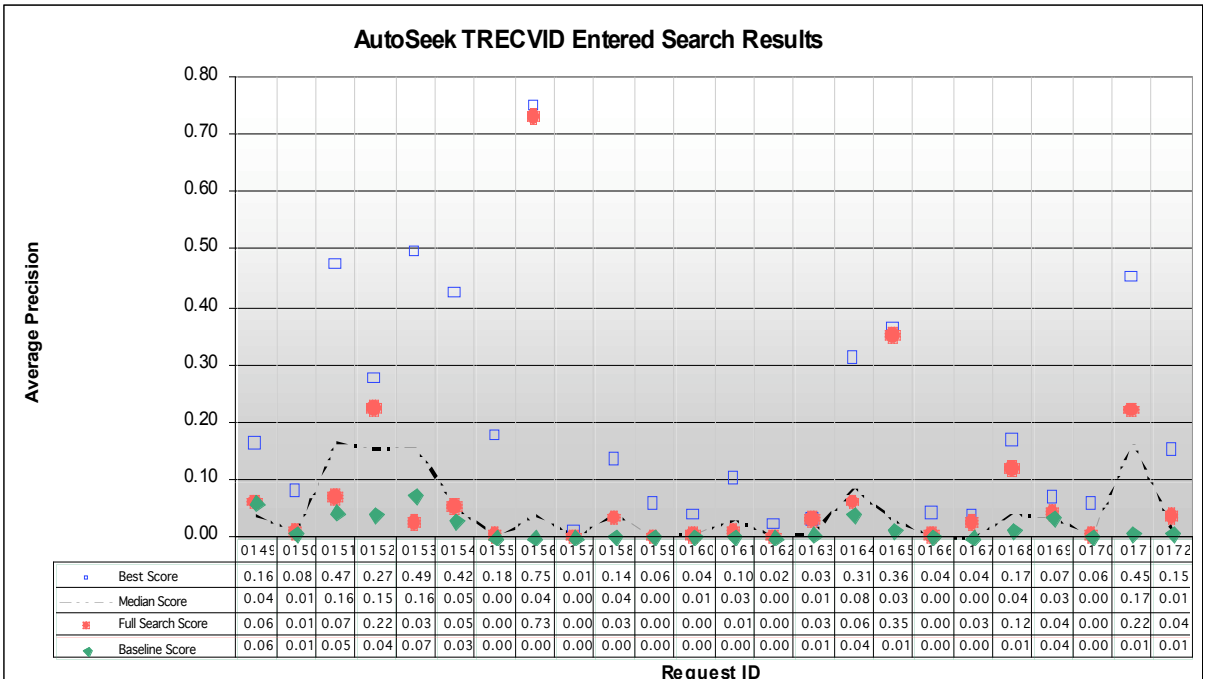


Figure 39. AutoSeek performance in TRECVID 2005 evaluation

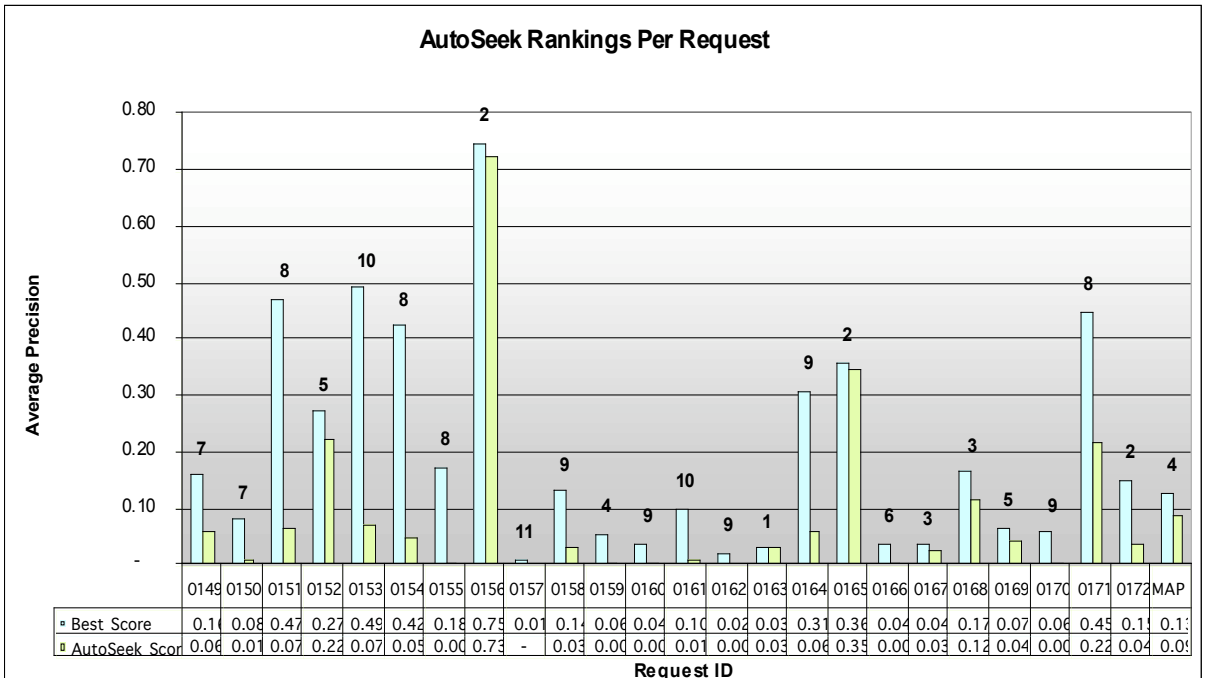
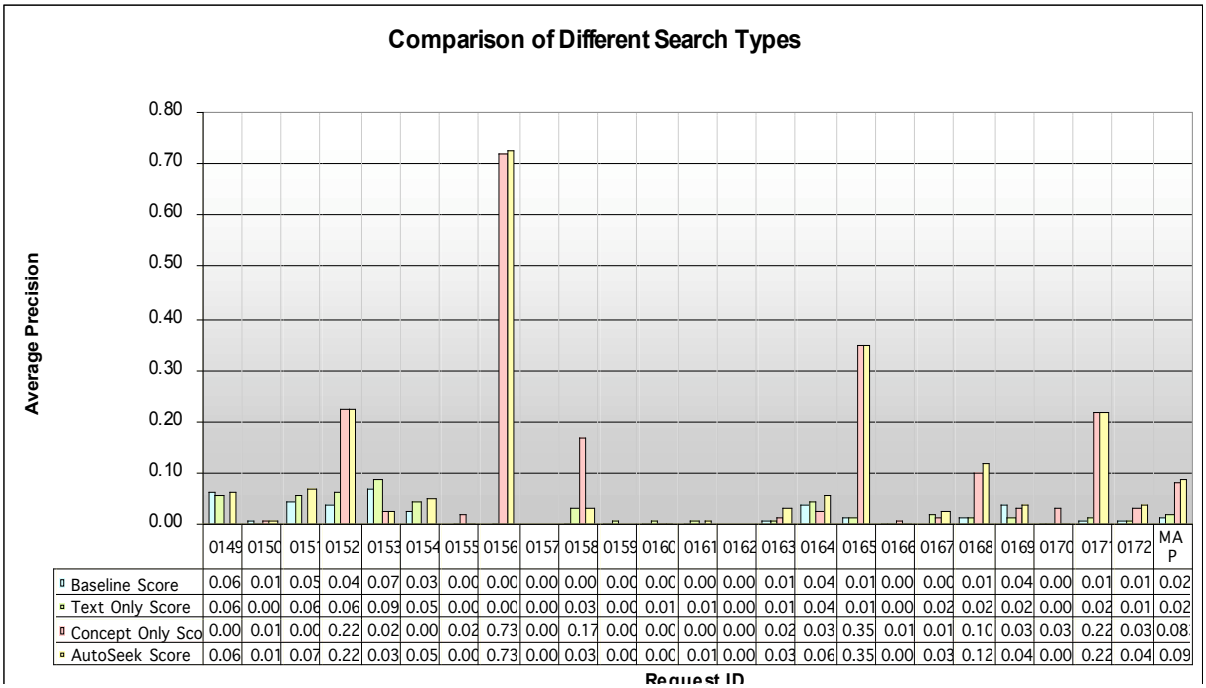


Figure 40. AutoSeek ranking compared to other teams participating in the Automatic search task



	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	MAP
Baseline Score	0.06	0.01	0.05	0.04	0.07	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.04	0.01	0.00	0.00	0.01	0.04	0.00	0.01	0.01	0.02
Text Only Score	0.06	0.00	0.06	0.06	0.09	0.05	0.00	0.00	0.00	0.03	0.00	0.01	0.01	0.00	0.01	0.04	0.01	0.00	0.02	0.02	0.02	0.00	0.02	0.01	0.02
Concept Only Sco	0.00	0.01	0.00	0.22	0.02	0.00	0.02	0.73	0.00	0.17	0.00	0.00	0.00	0.00	0.02	0.03	0.35	0.01	0.01	0.10	0.03	0.03	0.22	0.03	0.08
AutoSeek Score	0.06	0.01	0.07	0.22	0.03	0.05	0.00	0.73	0.00	0.03	0.00	0.00	0.01	0.00	0.03	0.06	0.35	0.00	0.03	0.12	0.04	0.00	0.22	0.04	0.09

Figure 41. Comparison of the text baseline, optimised text search, optimised concept search,and the final optimised AutoSeek search strategy.

5.2.1 Text Only Search

The optimised text search significantly outperformed the text baseline for 8 out of the 24 requests, and underperformed for 2 of the requests. The difference was not significant for the rest of the requests, though generally average precision was increased through use of the optimisation strategy. Overall, mean average precision increased from 0.0166 to 0.0215, showing the benefit of using the optimisation strategy for search through text. The 2 requests that did not benefit from optimisation, 0150 ("Find shots of Iyad Allawi, the former prime minister of Iraq") and 0169 ("Find shots of one or more tanks or other military vehicles"), are both general requests. We hypothesise that the optimisation for 0150 may not have worked well because of errors in the ASR transcript for the name "Iyad Allawi", which may have been corrected through the implicit expansion in LSI and have caused LSI to work very well.

5.2.2 Concept Only Search

Related semantic concept indices were found for 21 of the 24 requests. For 13 of the requests, more than one related index was found, as can be seen in Figure 42. In most cases, the concept selected for use in the final search is the concept that best matches the request. For example, for request 0159 there are many related semantic concept indices found, but the "George Bush Jr" index is finally selected as the index that best represents the query. This index is chosen because it is more specific (i.e. has a higher information content) than the other indices. Sometimes the final concept selection is less logical, for example in the case of request 0160, where shots with flames and smoke are required, but finally the aircraft index is chosen as the most specific index. This shows that the selection algorithm is not always correct. It is likely that the selection strategy would benefit from a subject detection, which identifies the main object of a request. Alternatively concept search might benefit from weighted combination of all the detected concepts.

ID	Request	Semantic Concept Indices Found	Concepts Used
0149	Find shots of Condoleeza Rice	none	none
0150	Find shots of Iyad Allawi, the former prime minister of Iraq	allawi	allawi
0151	Find shots of Omar Karami, the former prime minister of Lebannon	none	none
0152	Find shots of Hu Jintao, president of the People's Republic of China	hu jintao	hu jintao
0153	Find shots of Tony Blair	blair	blair
0154	Find shots of Mahmoud Abbas, also known as Abu Mazen, prime minister of the Palestinian Authority	none	none
0155	Find shots of a graphic map of Iraq, location of Bagdhad marked - not a weather map,	maps	maps
0156	Find shots of tennis players on the court - both players visible at same time	tennis, sports	tennis
0157	Find shots of people shaking hands	none	none
0158	Find shots of a helicopter in flight	aircraft, vehicle	aircraft
0159	Find shots of George W. Bush entering or leaving a vehicle (e.g., car, van, airplane, helicopter, etc) (he and vehicle both visible at the same time)	aircraft, vehicle, automobile, motorcycle, tank, truck, watercraft, george bush jr	george bush jr
0160	Find shots of something (e.g., vehicle, aircraft, building, etc) on fire with flames and smoke visible	aircraft, bicycle, building, automobile, fire, government building, motorcycle, tank, truck, vehicle, watercraft	aircraft
0161	Find shots of people with banners or signs	flag	flag
0162	Find shots of one or more people entering or leaving a building	building, government building	building, government building
0163	Find shots of a meeting with a large table and more than two people	meeting	meeting
0164	Find shots of a ship or boat	watercraft, vehicle	watercraft
0165	Find shots of basketball players on the court	basketball, sport	basketball
0166	Find shots of one or more palm trees	tree, vegetation	tree
0167	Find shots of an airplane taking off	vehicle, aircraft	aircraft
0168	Find shots of a road with one or more cars	road, car, vehicle	road, car
0169	Find shots of one or more tanks or other military vehicles	tank, vehicle	tank
0170	Find shots of a tall building (with more than 5 floors above the ground)	building, government building	building, government building
0171	Find shots of a goal being made in a soccer match	sport, soccer	soccer
0172	Find shots of an office setting, i.e., one or more desks/tables and one or more computers and one or more people	office	office

Figure 42. The related semantic concept indices found per request, and the final concept(s) used for search

The search on the selected concept index significantly outperforms the text baseline for 12 of the requests. In 7 cases there was no significant change in average precision, and in 2 cases average precision decreased. Mean average precision over all the requests (using an average precision of 0 for requests where no concept search was used) was 0.0837, which is higher than the baseline average precision of 0.0166. The requests for which the concept search resulted in decreased performance were requests 0153 and 0159. Concept detectors resulted in a decrease in average precision for these two requests despite the fact that they perfectly matched the selected concept index. The concept search for shots of Tony Blair performed badly because the "Tony Blair" index performed worse than a query on the words "Tony Blair", indicating that the index was of questionable quality. For the second request the "George Bush jr" index was selected, but average precision was low because the semantic concept indices gave the highest scores to shots of George Bush giving addresses and press conferences, while the request was for George Bush leaving a vehicle.

5.2.3 AutoSeek Combined Search

The final AutoSeek strategy resulted in a significant improvement in average precision over the text baseline for 12 of the 24 requests. In 10 cases the strategy did not result in a significant improvement, and in 2 cases the strategy significantly decreased the average precision. When comparing the final AutoSeek strategy to the best of the baseline, text, and concept searches, the average precision significantly exceeds the best average precision in 6 cases. The average precision matches the best average precision in 13 cases, and in 5 cases the use of one of the baseline, text or concept searches exceeds the average precision of the AutoSeek strategy.

ID	Request	Concepts Used	Concept Weight	Text Weight
0149	Find shots of Condoleeza Rice	none	0.00	1.00
0150	Find shots of Iyad Allawi, the former prime minister of Iraq	allawi	1.00	0.00
0151	Find shots of Omar Karami, the former prime minister of Lebannon	none	0.00	1.00
0152	Find shots of Hu Jintao, president of the People's Republic of China	hu jintao	1.00	0.00
0153	Find shots of Tony Blair	blair	1.00	0.00
0154	Find shots of Mahmoud Abbas, also known as Abu Mazen, prime minister of the Palestinian Authority	none	0.00	1.00
0155	Find shots of a graphic map of Iraq, location of Bagdhad marked - not a weather map,	maps	0.00	1.00
0156	Find shots of tennis players on the court - both players visible at same time	tennis	1.00	0.00
0157	Find shots of people shaking hands	none	0.00	1.00
0158	Find shots of a helicopter in flight	aircraft	0.00	1.00
0159	Find shots of George W. Bush entering or leaving a vehicle (e.g., car, van, airplane, helicopter, etc) (he and vehicle both visible at the same time)	george bush jr	1.00	0.00
0160	Find shots of something (e.g., vehicle, aircraft, building, etc) on fire with flames and smoke visible	aircraft	0.82	0.18
0161	Find shots of people with banners or signs	flag	0.00	1.00
0162	Find shots of one or more people entering or leaving a building	building, government building	0.82	0.18
0163	Find shots of a meeting with a large table and more than two people	meeting	0.50	0.50
0164	Find shots of a ship or boat	watercraft	0.82	0.18
0165	Find shots of basketball players on the court	basketball	1.00	0.00
0166	Find shots of one or more palm trees	tree	0.00	1.00
0167	Find shots of an airplane taking off	aircraft	0.82	0.18
0168	Find shots of a road with one or more cars	road, car	0.82	0.18
0169	Find shots of one or more tanks or other military vehicles	tank	0.82	0.18
0170	Find shots of a tall building (with more than 5 floors above the ground)	building, government building	0.50	0.50
0171	Find shots of a goal being made in a soccer match	soccer	1.00	0.00
0172	Find shots of an office setting, i.e., one or more desks/tables and one or more computers and one or more people	office	0.82	0.18

Figure 43. Requests and weightings used for the combination of text and concept search

The fact that the average precision is increased by combining text and concept results in 6 cases demonstrated that using a weighted combination of text and concepts can be a powerful way to boost average precision. In the 3 of the 5 cases where the final strategy does not provide the best result, AutoSeek still performs better than the baseline. In these cases AutoSeek would have benefited from weighting the concept search more highly, particularly for request 0158. In this case the “aircraft” concept search was given a weighting of 0. If a weighting of 1 had been employed, AutoSeek would have gained a better average precision for this request than any of the teams participating in the TRECVID evaluation. AutoSeek did not use this weighting because the information content measure considered the concept “helicopter” to have much higher information content than the concept “aircraft”, and therefore selected text rather than concept search to be the most appropriate measure.

CHAPTER 6 CONCLUSIONS, DISCUSSION, AND FUTURE WORK

6.1 Conclusions

We started with one question: *how can we perform automatic multimodal video search using only text as input?* We have outlined our solution in this thesis.

In Chapter 2 we reviewed the approaches taken by others for accomplishing automatic search, and found that all of their systems incorporate multimodal examples as an essential element. We have taken their research into account when developing our system, but it became apparent that an innovative approach was necessary to perform automatic search without the use of multimodal examples.

In Chapter 3 we outlined the design of AutoSeek, the system that has been designed and implemented to realise our goal of automatic multimodal video search using only text requests. We paid special attention to our first follow-up question: h*ow can we identify semantic concept detectors that are related to text?* Our solution was to link each semantic concept index to corresponding semantic concepts in the WordNet lexical resource. When a new search is entered AutoSeek extracts semantic concepts from text and locate them in WordNet, using the relationships within WordNet to identify connections between the text request and the available semantic concept indices.

We discussed the optimisation of the different system components and strategies in Chapter 4. In this chapter we included an investigation into our second follow-up question: *how can we combine results from heterogeneous types of searches into a single set of results?* We made use of rank-based fusion to overcome problems caused by different types of relevance scores used by different indices. We developed weighting strategies for merging results for different kinds of topics, incorporating four important weighting factors: the specificity of a request, the complexity of a request, the similarity of the concepts in request text to the available concept indices, and the information content of the concepts in both the concepts in the request text, and of any relevant indices.

In Chapter 5 we discussed the answer to our final question: *can an automatic multimodal video search engine that uses only text as input compete with engines that use multimodal examples as well as text?* The answer was yes. We submitted AutoSeek to external assessment in the TRECVID evaluations, and found that AutoSeek placed fourth overall as assessed by mean average precision, and ranked in the top three systems for 25% of the assessed requests. To the best of our knowledge, AutoSeek is the only participating automatic search system that uses only text as input, demonstrating that this strategy is feasible alternative to automatic search with examples.

From these findings we can conclude that AutoSeek provides a realistic solution for the problem of automatic search using only text as input. By using an approach grounded in the extraction of semantic concepts from both the multimedia database and the textual requests we are able to identify the semantic concept indices that are related to the textual request. By combining search using these indices with searches using text indices of ASR transcripts we are able to provide results that are competitive with the state-of-the-art in automatic multimodal search. This is good news for users. They will be able to perform a simple text search without spending time looking for multimodal examples of the video beforehand.

6.2 Discussion

A great advantage of AutoSeek is the generic approach taken to linking text and semantic concept indices, so that new semantic concept indices can be added at will. In fact, AutoSeek performed well even after increasing the number of indices in the system from the 28 used for development to the 88 used for evaluation. Given that Semantic Pathfinding allows us to quickly develop new concept detectors, this scalability is an important quality of our system. AutoSeek performs best when requests have related concept detectors present in the system, which implies that it will achieve better results as more semantic concept indices are introduced.

We expect that, as long as there are high-performance concept detectors available, AutoSeek will be extendable to genres of video other than the news broadcast genre for which it was developed, with the caveat that one important factor must be taken into account. A characteristic of the news video genre is annotative dialogue. Newscasters announce their subject, and often provide a running dialogue describing the main visual content of the stories that are being displayed, and therefore the combination of text with concepts plays in an important role in the AutoSeek strategy. Other video genres, such as cinematic productions or home movies, do not share this annotative characteristic. If AutoSeek is to be extended to other video genres, the optimisation outlined in Chapter 4 should be re-evaluated using the new data set.

One task in AutoSeek is the disambiguation of words with multiple meanings. We found that simply choosing the most common meaning of a word outperformed more sophisticated disambiguation measures for the TRECVID evaluation requests, but it is unclear whether the technique would perform well outside of the TRECVID evaluations. In a real-world situation it is realistic that there be at least a small amount of interaction. If AutoSeek is to be used in such a situation, for example through a web interface, we suggest that the user be allowed to perform manual disambiguation if they desire.

Time restrictions prevented us from implementing or investigating a number of measures that may have improved the performance of AutoSeek. One of these measures is blind relevance feedback, which we did implement, but were unable to investigate. Research by NUS PRIS indicates that this could aid retrieval. We also did not take into account the temporal mismatch between ASR text and the video. Other research indicates that inclusion of this aspect of news broadcast video could increase the efficiency of text search. Finally, the use of named entity extraction may be able to play a significant role in categorising different requests.

6.3 Suggestions for Future Research

We believe that one of the areas of automatic search that has the most potential is the combination of results from multiple semantic concept indices. In our research we chose to use only the most similar index, or if more than one index had the same high similarity, to weight each index equally. This is a simple approach, and there is room for improvement. Some requests have multiple corresponding indices, such as the request "Find shots of a meeting with a large table and one or more people". This request is related to the indices "meeting" and "table". AutoSeek utilises a search on "meeting" as it is the most specific concept, but it is feasible that the search would benefit from some sort of combination of the two concepts. This would require some logical analysis to distinguish between requests for "X and Y", "X or Y", and "X not Y". The investigation of this aspect of automatic search would require close coordination with the designers of semantic concept detectors, who have the best insight into the way that concepts are ranked.

We use a similarity measure to determine the best concept index to use in an automatic search. When applying this measure, each of the concept indices is assigned a score that indicates how close it is to the information request. By pre-selecting or prioritising the best matching concept indices as a user types a request, the measure could also be used to aid the user in an interactive search system. This would be of special use in a system that incorporates hundreds or thousands of concept detectors.

In the text retrieval field, statistically-based strategies are now commonly accepted as providing better results than knowledge based methods. AutoSeek, however, was developed using a very small set of development requests, and therefore employed a knowledge-based strategy for linking text to relevant indices. It is encouraging that AutoSeek still performed well for a number of searches. However, we believe that there is great potential in applying statistical methods to link between text input on the one hand, and the multimodal information in video on the other.

CHAPTER 7 REFERENCES

Abney, S. P. (1991). Parsing by Chunks. Principle-Based Parsing: Computation and Psycholinguistics. R. C. Berwick, S. P. Abney and C. Tenny. Dordrecht, Kluwer**:** 257-278.

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

Amir, A., M. Berg, S.-F. Chang, G. Iyengar, C.-Y. Lin, A. P. Natsev, C. Neti, H. Nock, M. Naphade, W. Hsu4, J. R. Smith, B. Tseng1, Y. Wu and D. Zhang (2003). IBM Research TRECVID-2003 Video Retrieval System. TRECVID 2003, Gaithersburg, MD.

Banerjee, S. and T. Pedersen (2003). Extended gloss overlaps as a measure of semantic relatedness. Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico.

Blott, S., O. Boydell, F. Camous, P. Ferguson, G. Gaughan, C. Gurrin, G. J. F. Jones, N. Murphy, N. O'Connor, A. F. Smeaton, B. Smyth and P. Wilkins (2004). Experiments in Terabyte Searching, Genomic Retrieval and Novelty Detection for TREC-2004. Thirteenth Text REtrieval Conference, Gaithersburg, MD.

Charniak, E. (1997). "Statistical Techniques for Natural Language Parsing." AI Magazine **18**(4): 33-44.

Chua, T.-S., S.-Y. Neo, K.-Y. Li, G. Wang, R. Shi, M. Zhao, H. Xu, Q. Tian, S. Gao and T. L. Nwe (2004). TRECVID 2004 Search and Feature Extraction Task by NUS PRIS. TRECVID 2004, Gaithersburg, MD.

Church, K. W. and P. Hanks (1990). "Word association norms, mutual information, and lexicography." Computational Linguistics **16**(1): 22-9.

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

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman (1990). Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science. **41:** 391.

Gauvain, J. L., L. Lamel and G. Adda (2002). The LIMSI Broadcast News transcription system. Speech communication: an interdisciplinary journal. **37:** 89-108.

Hauptmann, A., M.-Y. Chen, M. Christel, C. Huang, W.-H. Lin, T. Ng, N. Papernick, A. Velivelli, J. Yang, R. Yan, H. Yang and H. D. Wactlar (2004). Confounded Expectations: Informedia at TRECVID 2004. TRECVID 2004, Gaithersburg, MD.

Hiemstra, D. (2001). Using language models for information retrieval. Enschede, Neslia Paniculata; etc. |c 2001.**:** VIII, 164.

Ho, T. K., J. J. Hull and S. N. Srihari (1994). Decision Combination in Multiple Classifier Systems. IEEE transactions on pattern analysis and machine intelligence. **16:** 66-75.

Hollink, L., G. P. Nguyen, D. C. Koelma, A. T. Schreiber, M. Worring, P. Enser, Y. Kompatsiaris, N. E. O'Connor, A. F. Smeaton and A. W. M. Smeulders (2004). User strategies in video retrieval: a case study. Image and Video Retrieval. Third International Conference, CIVR 2004. Proceedings Lecture Notes in Comput. Sci. Vol.3115, Berlin, Springer-Verlag.

Ianeva, T., L. Boldareva, T. Westerveld, R. Cornacchia, D. Hiemstra and A. P. d. Vries (2004). Probabilistic Approaches to Video Retrieval: The Lowlands Team at TREC VID 2004. TRECVID 2004, Gaithersburg, MD.

Ide, N. and J. Véronis (1998). Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. Computational linguistics. **24:** 1-40.

Jiang, J. J. and D. W. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. Proceedings of International Conference Research on Computational Linguistics (ROCLING X), Taiwan.

Leacock, C. and M. Chodorow (1998). Combining local context and wordnet similarity for word sense identification. Wordnet an electronic lexical database. Cambridge, Mass., The MIT Press**:** XXII, 423.

Lin, D. and J. Shavlik (1998). An information-theoretic definition of similarity. Machine Learning. Fifteenth International Conference ICML'98, San Francisco, Morgan Kaufmann Publishers.

Marcus, M. P., B. Santorini and M. A. Marcinkiewicz (1993). "Building a Large Annotated Corpus of English: The Penn Treebank." Computational Linguistics **19**(2): 313-330.

Pedersen, T., S. Patwardhan and J. Michelizzi (2004). WordNet:Similarity - Measuring the Relatedness of Concepts. Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, CA.

Porter, M. F. (1980). An Algorithm for suffix stripping. Program: news of computers in British University libraries. **14:** 130.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. IJCAI 95. Fourteenth International Joint Conference on Artificial Intelligence, San Mateo, Morgan Kaufmann Publishers.
Robertson, S. E. (1995). Okapi at TREC-3. London, British Library Research and Development Department |c 1995.
Robertson, S. E., S. Walker and M. Beaulieu (2000). Experimentation as a way of life: Okapi at TREC. Information processing & management: libraries and information retrieval, systems and communication networks: an international journal. **36:** 95-108.
Salton, G. (1971). The SMART retrieval system: experiments in automatic document processing. Englewood Cliffs, NJ, Prentice-Hall.
Salton, G. and C. Buckley (1988). "Term-weighting approaches in automatic text retrieval." Information Processing & Management **24**(5): 513-23.
Salton, G. and C. Buckley (1990). "Improving Retrieval Performance by Relevance Feedback." Journal of the American Society for Information Science, **Vol. 41**(4): 288.
Santorini, B. (1990). "Part-of-Speech Tagging Guidelines for the Penn Treebank Project." Retrieved May, 2005, from http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/Penn-Treebank-Tagset.ps.
Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. International Conference on New Methods in Language Processing, Manchester, UK.
Smeulders, A. W. M. and M. Worring (2000). Content-Based Image Retrieval at the End of the Early Years. n. **22:** 1349.
Snoek, C. G. M. and M. Worring (2005). Multimodal Video Indexing. A Review of the State-of-the-art. Multimedia Tools and Applications. Boston, U.S.A, Kluwer Academic Publishers. **25:** 5-35.
Snoek, C. G. M., M. Worring, J.-M. Geusebroek, D. C. Koelma and F. J. Seinstra (2004). The MediaMill TRECVID 2004 Semantic Video Search Engine. TRECVID 2004, Gaithersburg, MD.
Snoek, C. G. M., M. Worring, J.-M. Geusebroek, D. C. Koelma, F. J. Seinstra and A. W. M. Smeulders (2005). The Semantic Pathfinder: A Unifying Architecture for Generic Indexing of Multimedia Archives, Universiteit van Amsterdam.
Triola, M. F. (2002). Essentials of statistics. Boston, Addison-Wesley.
Voorhees, E. M. (1999). Natural Language Processing and Information Retrieval. Lecture notes in computer science. **1714:** 32-48.
Voorhees, E. M. and D. K. Harman (2001). Common Evaluation Measures. The Text REtrieval Conference (TREC-2001) (10th, Gaithersburg, Maryland, November 13-16, 2001). NIST Special Publication.
Westerveld, T., T. Ianeva, L. Boldareva, A. P. d. Vries and D. Hiemstra (2003). Combining Information Sources for Video Retrieval. TRECVID 2003, Gaithersburg, MD, TRECVID.
Worring, M., G. P. Nguyen, L. Hollink, J. v. Gemert and D. C. Koelma (2003). Interactive search using indexing, fltering, browsing, and ranking. TRECVID 2003, Gaithersburg, MD, TRECVID.
Wu, Z. and M. Palmer (1994). Verb semantics and lexical selection. 32nd Annual Meeting of the Association for Computational Linguistics. Conference, San Francisco, Morgan Kaufmann Publishers.
Yan, R., A. Hauptmann and J. Yang (2004). Learning Query-Class Dependent Weights in Automatic Video Retrieval. ACM Multimedia 2004, New York, NY, ACM.

APPENDIX I DEFINITIONS

Below is a list of definitions of terms that occur in the text. The first time each of these terms occurs in the text, it is highlighted in **bold**.

ASR: Automatic Speech Recognition. A speech transcript of a video soundtrack.

Automatic Search: A search in which the user enters an initial information need, and all query selection, expansion and result ranking is done by the search system without any user intervention.

Chunking: The assignment of phrasal categorisations (for example, noun phrase, verb phrase, or prepositional phrase) to syntactically related groups of words.

Complex Request: Request that contains references to multiple objects. This is defined in AutoSeek as any request that contains multiple noun chunks.

Disambiguation: Choosing the intended meaning of a word from multiple possible meanings. For example, deciding whether the word "bank" in a sentence means the slope beside a body of water, or it means a financial institution that accepts deposits and channels the money into lending activities.

General Request: Request that does not contain a reference to a specific named person, object, or location. This is defined in AutoSeek as any request that does not contain a proper noun.

Ground Truth: Within the context of TRECVID, the ground truth for a information request is the subset of shots that have been explicitly judged to be either relevant or not relevant by a panel of human observers. These shots are selected from the top ranking results for the searches handed in for evaluation in TRECVID.

Holonym: A word that names the whole of which a given word is a part. "Hat" is a holonym for "brim" and "crown".

Hypernym: A word that is more generic than a given word.

Hyponym: A word that is more specific than a given word.

Index: A data structure that maps query terms to documents. For example, a text index usually maps words to documents that contain those words. An index for the feature "car" maps to documents according to the probability that they contain a car.

Interactive Search: A search in which the user iteratively interacts with a search system over an extended period of time, typically composing queries in different modalities and interactively helping the search system to rank results.

Inverse Document Frequency: A measure commonly used in text retrieval that describes how often a term occurs in a collection of many documents.

Latent Semantic Indexing: A text retrieval technique that uses a term-document matrix which describes the occurrences of terms in documents, with a number of dimensions equal to the number of unique terms in the matrix. The number of dimensions is then reduced that multiple terms are placed in the same location, resulting in implicit query expansion.

Low-Level Feature: A feature of video that can be directly derived from the data without any training – for example, a colour histogram.

Meronym: A word that names a part of a larger whole. "Brim" and "crown" are meronyms of "hat".

Multimodal Video Search: Video is a medium that contains multiple modalities: visual, audio, and temporal. We define a multimodal video search engine as a search engine that allows a user to incorporate more than one mode of video in a search request. A search engine that only search through ASR transcripts, for example, is not a multimodal video search engine.

Named Entity Extraction: A technique for identifying different categories of objects (for example locations, people, and organisations) as they occur in text. It is usually implemented using statistical methods and therefore requiring training upon a data set.

Polysemous: Having more than one meaning. In WordNet, a polysemous word is associated with more than one synset.

Query: A specification of a result to be calculated from an index.

Retrieval Technique: An individual search technique that has been implemented in the current TREC system. This includes keyword search, Latent Semantic Indexing, and concept search.

Search Strategy: A combination of retrieval techniques, as defined above, used in such a way as to allow the use of multiple techniques to perform an automated search.

Shot: Fundamental unit of video used for search. A shot is an uninterrupted, consecutive segment of video. Shots are generally between 1 and 30 seconds in length. Shots are detected through automatic shot boundary detection, a technique that is not always perfect but does offer a good approximation of true shot boundaries.

Simple Request: Request that contains references to a single object. This is defined in AutoSeek as any request that contains only one noun chunk.

Specific Request: Request that contains references to a named object or person. This is defined in AutoSeek as any request that contains a proper noun.

Stemming: The reduction of a word to a common root, for example by removing common suffixes.

Stopping: The removal of commonly occurring words from text.

Synset: The word used to describe a semantic concept as it occurs in WordNet. A synset consists of a group of words that describe the same semantic concept, usually accompanied by a short description.

Tagging: Assignment of part-of-speech categories (for example, noun, proper noun, adjective, verb) to words in natural language text.

Term Frequency: A measure commonly used in text retrieval that describes how often a term occurs in a single document.

TREC Answer: Video segment containing the visual concept(s) requested in the corresponding TREC topics. The number of relevant TREC answers for one TREC topics may be very large or very small.

MediaMill System: The system built by the MediaMill research group for the express purpose of participating in the TRECVID benchmark. It incorporates two large sets of video data fragments. The video fragments are associated with metadata indicating their date of recording, sequential order, calculated relevance to a number of different concepts, and speech transcripts. The TREC system also incorporates a number of different search strategies to navigate through the data. The TREC system referred to in this proposal does not refer to any system built for TREC benchmarks other than TRECVID.

TRECVID: Text REtrieval Conference Video Retrieval Evaluation. An annual two-day workshop organised by the American National Institute of Standards and Technology to encourage research in automatic segmentation, indexing, and content-based retrieval of video.

APPENDIX II PART-OF-SPEECH TAGS

Part-of-speech tags used in the TreeTagger are largely compliant with the Penn Treebank tags shown in Table 1. Schmid made some refinements to the verb tags. Specifically, the second letter of the verb part-of-speech tags distinguishes between "be" verbs (B), "have" verbs (H) and other verbs (V)[19].

Table 1. Part-of-speech tags used in Penn Treebank dataset(Santorini 1990)

Tag	Part-of-Speech	Tag	Part-of-Speech
CC	Coordinating conjunction	PP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PP	Personal pronoun	WRB	Wh-adverb

[19] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/