Exploiting inter-conceptual relationships to boost SVM classification.

Gert-Jan Poulisse

Enschede, August 2007

Table of contents

Abstract

Concept detection is the process of extracting semantic meaning from data. Video data is a popular choice on which to operate, as there is a lot of visual, audio, and textual information to index and search. Ultimately one would like to develop a set of semantic concepts that spans the search space, but this requires defining thousands of concepts. In order to detect such a copious amount of concepts, generic concept detectors have to be employed. There is a continuous drive in research to discover better ways to perform generic concept recognition.

This thesis starts with a literature overview, surveying past and future trends in concept detection. Past classification systems were often rule-based systems that made use of specific domain knowledge to perform their tasks. While functional, these systems could not readily be extended beyond their domain. State of the art classification systems on the other hand, use statistical models, in the form of Support Vector Machine classifiers, to recognize an unbounded set of concepts.

The initial thrust of this investigation was to examine the potential of using SVM classifiers to detect an abstract concept, such as 'happiness', by relating simpler, indicative concepts. This proved infeasible, and the focus of this research became to improve weak classifiers by exploiting the knowledge of more discernable, related classes. Three techniques were developed in this study that did this, each applicable to a different type of inter-conceptual relationship. This thesis aims to assess the performance and the associated constraints of these developed techniques.

The Sibling-confusion removal and Ancestor boosting techniques require an ontology, a tree-like structure that models semantic relationships between concepts by linking relationships in a hierarchy. The Sibling-confusion removal technique attempts to improve detector performance by removing false positives caused by similarities between sibling concepts. The Ancestor boosting technique aims to improve poorly performing child detectors by leveraging the functionality of their more powerful ancestor concept detectors.

The final technique used a statistical method, the chi-square test, to identify concepts in the dataset that frequently appeared simultaneously. Concept recognition was improved by combining the outputs from related detectors to recognize a single concept.

In the course of the experiments, a number of hidden constraints for each technique became apparent and explain the results thus obtained. Sibling-confusion removal proved to be a worthwhile technique when the ontology provides a concept grouping, which is semantically related, closed, and for which only one concept is valid in each shot. Ancestor boosting appears to be a promising technique, as evinced by substantial increase in detector performance for some concepts in the dataset. For Ancestor boosting to work successfully, however, it is necessary that ancestor and child concepts be tightly linked semantically and that ancestor detectors perform robustly. Chi-square boosting is a powerful technique, as it identifies concept relationships that are not immediately obvious from their semantic definitions. Most of the discovered concept relationships may be used to produce improved concept detectors.

The MediaMill Challenge dataset, consisting of 101 semantic concepts, was used to test the effectiveness of each technique. The mean average precision (MAP) of each original concept detector was compared against the mean average precision score of the revised concept detectors. In the Sibling-confusion removal experiment, 30 out of 64 distinct concepts had improved MAP scores, while 17 out of 61 distinct concepts had improved MAP scores in the Ancestor boosting experiment. The Chi-square boosting experiment had an improvement in 22 out of 36 concepts.

Acknowledgement

I wish to recognize the considerable assistance afforded to the conduct of this research and preparation of the manuscript by Maarten Fokkinga, CS Department, Twente University. His particular ability to stimulate and direct my curiosity, essential for charting unknown SVM database classification waters, is most gratefully acknowledged. Thanks are also due to Robin Aly for his useful discussions when in search for solutions.

GJP,
Enschede, August 2007

Chapter 1

Introduction

When a person goes to a public library to look for a book, he first goes to the card catalogue and looks for a book in the category he desires within the catalogue. Thus he has a fuzzy idea of what he is looking for, in the sense that he knows a few key words to describe it. The catalogue is ordered so that the keywords will narrow his search until he finds what he is looking for. This assumes however, that each book has been previously indexed and placed within the catalogue. If one extends the metaphor to searching for video footage, one realizes that the audio, video, and text streams that make up the video recording must also somehow be indexed. This is complicated however by that fact that a human would index by providing a textual summation of the content. To a computer however, the video stream is merely a sequence of images, with each image being a set of colored points. This is known as the semantic gap.

Nonetheless, it is possible to train a computer to recognize low-level features, such as the colors of an image, and associate them with concepts. The implicit loss of data associated with indexing, plus the ill-defined nature of semantic concepts means that this process introduces error. In addition, most sophisticated concepts can only be recognized by the presence of simpler concepts. For example, a car-chase scene could only be recognized if previous classifiers have recognized multiple cars following each other at high speeds. The art then, is to map low-level features to a concept vocabulary that covers the human language that minimizes error and provide maximum concept coverage.

Early research focused on combining multimodal feature extractors in various ad-hoc approaches to identify specific concepts. Unfortunately, this does not scale well, as the combination of feature extractors is case specific. Thus one cannot use the same combination of feature extractors to recognize a different concept. More recent research, such as the TRECVID high-level feature extraction task, focuses on implementing a framework of generic concept detectors to define a vocabulary that spans the human language. This task can be done by defining each concept as a unique blend of constituent features, or defines the concept to be identified in terms of other concepts. The challenge is to find an optimal combination of feature vectors and classifiers; and concept detection to date remains wide open to further research.

1.1 Goals

The initial aim of this research was to utilize the semantic relationships between some basic concepts to develop a concept detector capable of recognizing an abstract concept like ‘happiness’ in a dataset. Such a detector proved infeasible, and the focus of this research became to develop methods to improve weak classifiers by exploiting the knowledge of more discernable, related concepts.

The hypothesis is that concept detectors can be improved for concepts, which are semantically or statistically related, by making use of the additional information these relationships provide.

1.2 Approach

An initial literature study was performed in order to discern past and current trends in research with respect to semantic concept detection. This study revealed that SVM classifiers were the most promising classifiers to date, and so it was decided to detect an abstract concept, such as 'happiness', by relating simpler, indicative concepts using SVM. A preliminary investigation showed this was not possible given the limitations of the available datasets. Instead, three techniques, inspired by the previously discovered literature, were developed to validate the hypothesis.

An ontology was created that models semantic relationships between concepts by linking relationships in a hierarchy. Two techniques, Sibling-confusion removal and Ancestor boosting utilized this ontology. The Sibling-confusion removal technique attempts to improve detector performance by removing false positives caused by similarities between sibling concepts. The Ancestor boosting technique aims to improve poorly performing child detectors by leveraging the functionality of their more powerful ancestor concept detectors.

A final technique was developed that used the Chi-square test to identify concepts that frequently appeared simultaneously. Concept recognition was improved by combining the outputs from related detectors to recognize a single concept.

The mean average precision was computed for all the concepts in the dataset, before and after the application of these techniques. The increase in mean average precision scores for some concepts serves to confirm the hypothesis.

1.3 Organization of this paper

This paper is organized as follows:

Chapter 2, *Concept detection in literature*, discusses past and present research efforts in concept detection. The aim of this survey was to present various concept detection techniques, their comparative merits, and the applicability of these techniques in detecting a wider set of concepts. The trend in research is moving away from knowledge-based systems to generic concept classifiers afforded by Support Vector Machines.

Chapter 3, *Methodology*, describes various practical issues related to the choice of supervised learner, annotation software, and dataset. In conjunction with some preliminary findings, these choices affected a change in the method of approach.

Chapter 4, *Inter-conceptual boosting experiments*, describes three techniques which aim to improve concept detector performance by using knowledge of the semantic and statistical concept relationships in a dataset. The Sibling-confusion removal technique attempts to improve detector performance by removing false positives caused by similarities between sibling concepts. The Ancestor boosting technique aims to improve poorly performing child detectors by leveraging the functionality of their more powerful ancestor concept detectors. In Chi-square boosting, concept recognition was improved by combining the outputs from related detectors to recognize a single concept. These techniques were evaluated on the MediaMill dataset, and their results are analyzed.

Chapter 5, *Conclusion*, discusses the conclusions of the paper and suggests further refinements in the techniques applied.

Annex 1, *SVM Theory*, presents a summarized mathematical background of Support Vector Machines and briefly introduces the parameter settings that influence the development of a SVM model.

Annex 2, *Sibling-confusion removal results*, presents the results obtained using the Sibling-confusion removal technique developed in Chapter 4.

Annex 3, *Ancestor boosting results*, presents the results obtained using the Ancestor boosting technique developed in Chapter 4.

Annex 4, *Chi-square boosting results*, presents the results obtained using the Chi-square boosting technique developed in Chapter 4.

Chapter 2

Concept detection in literature

This chapter presents a review of a selection of papers, showing a chronological progression, documenting the progress made in research in the field of concept detection. The aim of this survey was to discover the various concept detection techniques used in research, their comparative merits, and the applicability of these techniques in detecting a wider set of concepts. Some statistical classification systems and techniques presented here provide the basis for work done in chapter 4.

2.1 Related Works

For video data, there are three types input streams, the audio, video, and text (transcriptions of the words spoken in the segment). Feature extraction then, is the action of determining the characteristics for of the video fragment, in any of the three modalities, to detect some sort of concept. Some examples of features are: color histograms (indicative of the colors in the video), edge orientation histograms (represents the various edges of shapes in the video), Mel-frequency cepstrum coefficients (indicative of the rate of change of the audio), or word frequency (the number of occurrences of various words of (spoken) text). The process of combining these features in order to recognize a particular semantic concept is known as classification, or fusion.

Early research on semantic concept meta-classification examined ad-hoc rule based domain knowledge schemes, Bayesian classifiers (BN), neural network classification (NN), Gaussian mixture models(GMM), modeling via ontologies, Support Vector Machines(SVM), or Hidden Markov models(HMM). The various approaches are either statistical in nature (Bayesian, GMM, HMM, SVM), or are knowledge based, using knowledge of the domain (rule based, modeling using ontologies).

The drive to create a framework of detectors capable of recognizing generic concepts precludes the use of domain-based classifiers and is the reason for the trend towards statistical methods in research. This is not to say that domain based meta-classifiers perform poorly. Often they make use of insights, such as the dependency between two concepts, that short circuit the whole machine learning process of statistical methods, saving much development time. Their failure is in being unable to function properly for events outside their specific domain. For example, the highlights detection of Babaguchi [7] (discussed in 2.2) would fail for say, Formula 1. The reasoning behind using statistical classification is that enough features are considered that a concept is identified correctly, no matter the domain. In addition, the detection process as a whole is more robust, as more features are considered, and as such the error contribution per individual feature is lessened.

Of course, there are various problems for statistical classifiers. In general, increasing the feature set improves the accuracy of the performance, but also leads to over-training and the curse of dimensionality. The curse of dimensionality occurs as a result of an increasing feature space, when the increase in dimensions causes the distance between objects to become increasingly similar, and hence the objects become harder to distinguish and thus classify [63]. Likewise the time needed for the actual machine learning process increases exponentially with the amount of features under consideration. Knowledge based classifiers simply avoid this by reducing the feature set by using knowledge of the domain. Another consideration is that the SVM fusion can

be performed in various ways, each with its own tradeoffs between extensibility, robustness, and the time spent on learning the semantic concepts.

Recent research shows a distinct preference for SVM classification, because it gives superior performance over other statistical approaches and because it is robust against overtraining and the curse of dimensionality[30]. For further understanding of the mathematical reasons for this, see Annex 1. Although some domain approaches are still attempted, statistical classification is now the trend. In the field of domain knowledge classifications, the work done on ontologies is a recent innovation. However, in this instance ontologies are often deployed on top of a generic classifier (such as SVM's) to produce a hybrid attempting to incorporate domain knowledge on top of a statistical classifier.

Early research focused on developing ad-hoc concept recognition systems. They were rule-based, and operated on a fixed domain. These are reviewed in section 2.2. The desire to detect a much larger concept set led to the development of statistical concept detection methods, and are discussed in 2.3.

2.2 Domain Based Classification

Some of the first multimedia information retrieval systems to be developed investigated sports video. The system by Babaguchi and Nitta[7] was designed to analyze sports video, specifically baseball and American football, and determine the presence of semantic concepts such as highlights, live plays, crowd cheering, and the type of scene currently playing. Highlights were detected by examining the text stream for domain specific keyword phrases such as "touchdown" and then finding the corresponding time interval in the video stream. Crowd cheering was determined by the short time energy feature of the audio stream. Using the idea that crowd cheering was indicative of a highlight moments, a more sophisticated detector was developed by excluding highlights without cheering [7]. This system is an example of how specialized domain knowledge can readily provide a successful solution to identifying a specific semantic concept, such as highlights. The extensibility of the system is however open to question.

Haering et al [20], however, did develop a system seeking extensibility. The prototype system was designed to detect animal hunts in wildlife video, which is a complex semantic event. The promise of extensibility comes from the development of a modular, tiered system to allow easy redeployment for the detection of different semantic events. The first tier of the system extracted basic color, texture, and motion features, moving object blobs as well as shot boundary locations. Using these features, a neural network determined the class of the object under consideration. Nine of them were specific animals, five non-animals corresponding to rocks, sky/clouds, grass, trees, and a final unknown class. The third, and highest, tier of the system, in essence the meta-classifier, used domain specific rules to detect semantic events based on a combination of mid-level object descriptors spatially or temporally ordered based on the features from the first level.

Despite using domain specific knowledge, the system is readily extensible since the first tier is entirely domain independent; they are low-level image features after all, as is the second tier. The neural network classifier needs to be retrained to recognize additional objects, to be extended, but that cannot be avoided. Only the third tier would have to be adapted to a new domain, since the rule-based inferences of the first and second tier features would be different [20]. Arguably, a statistical classifier could replace the third tier, but at the cost of time spent on the machine learning process. Accommodating the rule-based semantic events to an increase in the number of objects could get exponentially complex.

Returning to concept detection in sports video, Xu [64] developed a somewhat domain (team-sport) independent system capable of handling semantic events which do not have significant audio/video features, such as when players are given yellow/red cards in soccer. He argues that most audio/video patterns are insufficiently distinct to recognize such semantic events. Likewise he argues that his system is readily extensible. His approach is to detect generic video concepts, using Hidden Markov Models (HMM), from the audio/video stream, such as shot category, focal distance, special view category, field zone, camera motion direction, and motion activity. Another HMM classifier is used to detect the transition between such events in the video stream. Domain dependencies are introduced in the form of external text streams detailing for instance game rules (important for field type and match duration), player names (facilitates text analysis), and event types (linking event types with AV patterns detected by the HMM), to detect more detailed semantic concepts. The assumption is that only noteworthy events are included in a match report.

Sports events defined in a text stream are aligned against previously detected generic video events, which constitute another classification problem. Xu compares three fusion methods, a rule-based scheme, a probabilistic aggregation scheme, and one using Bayesian inference that perform this alignment. The rule based scheme aligns text events within a temporal window based on the number of matches between text events and the externally provided, domain specific event model. Since text and video stream events are usually misaligned by some offset, the aggregation method models a semantic event as the combined probability of the event occurring in one stream and the probability of the event occurring in the other stream, offset by some margin. The margin is determined by gradient descent during a training phase. The last fusion method, Bayesian Inference, considers whether an event occurring in one stream occurs within a fixed offset in the other stream [64].

In terms of precision and recall, rule-based fusion gives the best performance, with Bayesian inference only mildly less accurate. Aggregation is the poorest performer [64]. All results have precision and recall above 84%. Xu attributes this discrepancy to sensitivity of the aggregation method and the large randomness in time offsets. The rule-based method benefits from using the additional detail possibly in the text stream and as such can correctly identify more events. The Bayesian Inference is unable to do so, and hence performs slightly worse. [64]

Xu's system is a reasonably generic system for sports, with good precision and recall. There is support for extending the system, the only caveat being that every sport needs external, domain specific parameters. Xu argues that this data can often automatically be retrieved and parsed, whereas event models are non-volatile after construction. Provided there is some operator assistance to develop these models, the system can support a large number of sports. For increased performance, rule based fusion could be employed, requiring additional operator assistance to develop these alignment rules. For a slight drop in performance, but no requirement for human intervention, Baysian Inference would suffice.

The chronological progression of papers presented here illustrate the advances in concept detection. Early systems were ad-hoc attempts to perform some basic highlight detection [7] or animal recognition [20]. More sophisticated systems attempted to move beyond the fixed domain constraints of knowledge-based systems. Xu's sports detection system [64] does this by utilizing a collection of rules necessary to recognize the semantic events specific to a domain. He contends that these event-rules can easily be generated for each new domain. Nonetheless, this ultimately seems too impractical an approach for a system that wishes to detect generic concepts.

2.3 Statistical Based Classification

The papers reviewed in this section perform classification using statistical methods, such as Hidden Markov models [1,2], Gaussian mixture models [1], or Support Vector Machines [1, 26, 30, 47, 55]. They are of interest because chronologically early papers contrast various classification methods, and determine that SVM gives superior classification performance [1, 26, 30, 47]. Later papers describe various methods to further improve SVM performance [26, 55, 62].

Alatan's system [2] aims to detect dialogue scenes in video, and uses Hidden Markov Models (HMM) as the classifier. The dialogue scene, or story, is defined as a set of consecutive shots that make up a meaningful and distinct part of a whole story. An example of this would be a scene from a news broadcast. This would contain the shots of the news anchor introducing a news item, the news item itself, and possibly any concluding remarks made back in the studio. A scene is always present in video, irrespective the genre, and thus scene detection results in the partitioning of the video into semantically meaningful, logical units. What makes scene detection difficult is the absence of a fixed format to a scene. Care must be taken to neither miss shots that should be part of a scene, nor to accidentally subdivide a scene because of intermittent shots that break the visual flow (such as a close up) and yet are semantically relevant to the whole.

Alatan models a scene as consisting of three elements, people, conversation, and a location. People are detected using face detection, while audio is classified as either music, speech or silence. Shifts in location are detected by analyzing the histograms of several consecutive shots. The results of each detector are then used as inputs of an HMM to detect, and classify, scenes as either establishing, dialogue, or transitional, the three types most commonly used by film directors. He argues for HMM over rule-based, deterministic methods because HMM allow for random behavior, such as extraneous shots within a scene, as one might expect when analyzing video without any prior knowledge of the content. [2]

The use of a HMM classifier avoids the domain dependence of rule based classifiers, and can readily be made more robust by adding more classifiers as inputs. This would not however, require the alteration of the pre-existing classifiers. Likewise more semantic inferences, for example more distinct scene types, could be made by extending the output classification set of the particular HMM, although as with adding additional classifiers as inputs, each alteration requires the retraining of the HMM.

Snoek and Worring [47] also developed a system for use in the news and sports (soccer) domain. They propose a framework, called TIME, which is a multimodal approach to tackle the problems of context and time-synchronization common to these domains. This framework is evaluated using three statistical classifiers, C4.5 decision trees, Maximum Entropy, and SVMs. The choice for statistical classifiers was made in order to provide for a robust performance in domains such as soccer, where events are sparse, context dependent, and unpredictable.

Low level feature extractors operating on the video stream detect various multimodal events, such as the camera shot type, microphone shot, text shots, panning camera, speech, speech excitement, motion intensity, close-up, goal related keywords. These features have additional context information added by temporally relating them using the labels {precedes, meets, overlaps, starts, during, finishes, equals}, thus producing events. Events are assumed to always have at least a time distance of T1, due to noise. If events are separated by an interval of T2, then they are assumed to have no temporal relationship with each other. Semantic concepts can thus be modeled as a combination of time ordered features within a certain interval, as determined by a classifier.

C4.5 decision trees place these events into a binary tree based on a gain ratio determined at training time. Each concept is a leaf node in the tree, and the time-ordered events form decision nodes higher up in the tree. The more important the event is to the classification task, the higher it is in the tree [47]. A Maximum entropy (MaxEnt) classifier estimates the conditional distribution of a concept in a video, given certain constraints. These constraints are features, whose values are determined from the training set [47,68].

In the soccer domain, where concepts such {goal, yellow card, substitutions} concepts were looked for, C4.5 decision trees gave the poorest performance. MaxEnt and SVM detected all semantic concepts equally well. What differentiated them was that the SVM classifier required considerably less training time than the MaxEnt algorithm to achieve this result [47]. In the news domain, where concepts such as {reporting anchor, monologue, split-view interview, and weather-report} were sought, the SVM classifier outperformed the C4.5 and MaxEnt classifiers, both of whom performed similarly. In an additional experiment to test the effectiveness of the TIME framework, SVM based classification on the news domain was performed with temporal relations enabled and disabled. For most semantic concepts, the additional information provided by the TIME framework yielded increased performance, except for the weather report, where results were comparable [47].

The Time framework demonstrates that it is possible to add additional contextual information, in this case a temporal ordering, to low level concepts. This additional information results in better performance of the classifier than when it is not provided. This Time framework also suggests that SVM classifiers outperform C4.5 decision trees and MaxEnt classifiers over two different domains, and one could speculate that this would also apply for other domains.

One of the earliest applications of a SVM classifier was a 2002 system from Carnegie Mellon which integrated a video camera and two microphones in a tape-recorder like system. The video camera provided input to two face recognition detectors, while the microphones had feature detectors checking for speech identification by similarity and pitch. The purpose of the system was to remind the user of the last conversation, if any, had with a dialogue partner. The results clearly demonstrated that the individual detector results, or a summation of their results, resulted in a significantly poorer performance than when their outputs were fused using an SVM (late fusion) classifier [30].

IBM [1] has also focused research on multimedia retrieval. Rather than attempting a domain specific application, their system was explicitly designed to explore concept detection and the performance of various fusion schemes. Their system used machine learning over low level features on the audio, visual and text channels to determine the most effective model for various concepts. For all fusion methods however, late fusion was employed to combine unimodal features concept classification. Statistical classification, through the use of Support Vector Machines, and probabilistic modeling approaches, such as Gaussian Mixture Models (GMM), HMM, and Bayesian networks, were investigated. GMM and SVM performance was compared for visual features, while GMM and HMM performance was compared for fusion of audio features. The resultant concepts were considered unimodal, or atomic concepts. An investigation was made into the appropriate fusion model for high level concepts; concepts which can only be inferred by the presence of other concepts and low level features and are generally multimodal in nature. For this task, the performance of Bayesian networks was compared with a SVM classifier [1]. The video footage from the TREC 2001 corpus was used for evaluation.

For unimodal classification of visual features, which examined SVM versus GMM performance for visual concepts such as {outdoors, sky, rocket, fire/smoke}, SVM classifier performance considerably outperformed GMM accuracy, with over 90% precision for most of the

recall range. Even with a small training set, SVM classifiers provided a reasonably accurate detection performance. [1]

For unimodal classification of audio features, which examined HMM versus GMM performance for the classification of {rocket engine explosion, music, speech, speech + music}, HMM precision outperformed GMM's over all recall values [1]. These concepts were then used in an additional experiment examining the best fusion method to detect the semantic concept, 'rocket launch'. Explicit fusion used the classification results from the previous unimodal classifiers as inputs into a Bayesian network to detect this concept. Implicit fusion uses the following function to generate a score for each concept:

$F(c_i) = f(c_1 \ldots c_n) = Score(c_i)/(\Sigma(Score(c_1 \ldots c_n))$ *where* $Score(c_i)$ *is the unimodal score for each concept in a shot.*

Each concept is normalized by the sum of all the scores for concepts present in a given shot. For this particular concept, implicit fusion outperformed explicit fusion over all recall values. [1]

Although implicit outperformed explicit fusion, I would question the validity of this classifier. Implicit fusion is discriminative in nature as it boosts the most dominant audio cue. In this particular instance, the semantic concept of a 'rocket launch', is detected given the more basic concept of a rocket engine explosion. Since there is only a single concept which positively contributes to the 'rocket launch' event, implicit fusion, which discriminates between various audio cues, will naturally give a good score. It is likely that this method would fail on high-level semantic concepts, which might be made up of multiple distinct audio cues, unlike explicit fusion.

The experiment examined semantic classification of the 'rocket launch' event over multiple modalities. Recall the visual unimodal classifiers detected concepts such as {outdoors, sky, rocket, fire / smoke} while the audio unimodal classifier detected the 'rocket engine' event. These concepts were used as inputs for a Bayesian classifier in order to detect the rocket launch event. The SVM classifier instead took visual concepts, {outdoors, sky, rocket, fire/smoke}, audio concepts {rocket engine explosion, music, speech, speech + music}, and the occurrence of the word 'rocket launch' from automatic speech recognition as inputs to classify the rocket launch event. Both gave comparative precision over the recall curve, and outperformed any unimodal classifiers alone. The SVM classifier also outperformed the Bayesian classifier [1].

The research performed a comparative analysis of various semantic classifiers. Gaussian mixture models clearly were less suitable than Hidden Markov models (HMM) or Support Vector Machines (SVM) for unimodal classifiers. Possibly further experimentation could have successfully demonstrated, however, the effectiveness of implicit over explicit fusion (Bayesian Network). Both multimodal Bayesian Networks and Support Vector Machines (SVMs) performed better than their unimodal counterparts, and had comparative precision and recall in detecting the rocket launch event.

Iyengar[26] et al, 2003, also from IBM, extended the work from [1]. Using the same setup of basic concepts, they showed that a SVM classifier outperformed a Bayesian network for the detection of the rocket launch semantic concept, although not by too large a margin. Of additional interest is the questions raised over the requirements of an extensible, generic semantic concept detection system.

Apart from the obvious challenge how to make the system as accurate as possible, the research also addressed coverage, which is a measure for how many concepts a multimedia retrieval system can define reliably. Their Discriminitative Model Fusion (DMF), actually a multimodal SVM based classifier, is considered more accurate because it equaled or

outperformed the best unimodal specialized detectors for concepts in the TREC 2002 corpus. The DMF system was tightly coupled to an annotation system, allowing for the quick addition of arbitrary semantic concepts given some sample shots. Six arbitrary concepts were thus defined and DMF gave significantly better results than specialized detectors created for the occasion. Thus the system demonstrated its easy extendibility to incorporate additional concepts. Open questions were left regarding what constituted an optimal number of basis detectors, the total discriminatory capacity of the DMF framework given such a basis set of detectors, and the minimum required training set size per concept [26]. The research considered several key issues regarding classification in relation to developing a generic, easily extensible, robust semantic concept detection engine.

In a similar investigation into classifier performance by IBM [55], the thrust was more on a comparison of early fusion and a late fusion method (termed normalized ensemble fusion) that retained some decision making control over classifier combinations. The argument was that, although early fusion preserves all information but suffers from some practical constraints, such as a limit in the numbers of training examples, a limit in computational resources for training, and the risk of over fitting the data. An alternative, late fusion method was developed. Early fusion was performed by merging the feature sets, before performing training to create a classifier. Normalized ensemble fusion consisted of normalizing the output of individual SVM feature classifiers, via rank, range, or Gaussian normalization. Per semantic concept, the most high performing and complementary set of feature classifiers was chosen for aggregation by a combiner function. The combiner function considered minimum, maximum, average, product, inverse entropy, and inverse variance combinations to arrive at a classifier for a concept.

As a final experiment, all the SVMs that made up a concept classifier were evaluated using varying kernels. Kernels are functions which transform inputs into a higher dimensional space, and are further explained in Annex 1. When evaluating these combinations against the validating set, the resultant classifier was chosen that most confidently classified their samples, as measured by a samples' distance from the separating hyper plane. Thus in normalized ensemble fusion, the classifier was trained by the most confidently classified concept, using a feature selection set that gave the best average precision. This fusion method was the best performing system at TREC 2002. It also outperformed early fusion, which had an average precision of 0.5896 versus 0.71 [55]. The research developed a strong late fusion method which combines the power of SVM classifiers with a semantic concept-specific soft decision combinatory function and a powerful late fusion concept detector.

Also originating from the IBM labs is the idea to enhance the semantic classifier by using additional information provided by a hierarchical tree of related semantic concepts, in other words, an ontology [62]. In statistical modeling the assumption is that a high correlation in the feature space will produce similar classification output, although there might not actually be a relation between the semantic concepts. Thus, especially in the case where there are few training examples, unreliable classifications are the result. For example, the concept 'Desert', of which there were only 17 instances, in a data set of 9852, was only correctly detected with an average precision of 0.06. In the same dataset, 'Outdoors', with 2473 occurrences, was detected with an average precision of 0.58. This illustrates how an insufficient training set leads to a poor classifier.

The research developed two algorithms to enhance classifier performance. When training the classifier of a child concept, the confidence scores of the more reliable ancestor classifiers are considered and influence a child concept detector's score. The extent of the boosting-influence of the ancestors on the child node is related to their confidence score distributions. If a child and its ancestor have a similar confidence score distribution, they are likely to tightly relate too on a

semantic level, and the ensuing boost in confidence score becomes greater too. Boosting is done for all ancestors of a child concept. The other algorithm considers the confusion factor, which is defined as the probability of misclassifying data into one semantic-class, while in reality the data belongs to a mutually exclusive different class. Data points are checked to see if they have not been placed in the wrong semantic class, and the confidence scores are updated accordingly. Each semantic concept was initially modeled using SVMs. The resultant output classifications are screened for confusion and boosted according to the semantic relations in the ontology [62]. When tested using the TRECVid-2003 data, this ontology-based classifier outperformed the previously developed Discriminitative Model Fusion method [26] by 6% over 17 concepts, and by 23% over 64 concepts. It bettered the best unimodal classifiers by 42% [62].

The research is of significant importance as it demonstrates the next evolutionary step of semantic machine learning, which relies on semantic relationships, as evinced in the use of language ontology. Of course, this system too was built on top of SVM classifiers, but the addition of ontology was key in outperforming plain SVM meta-classifiers, such as the DMF system. It also compensates for a weakness of SVM classifiers, when there are simply too little training data from which to derive an adequate classification model.

The papers presented here describe the chronological progression of research into statistical classifiers. Early papers compared and contrast various classifiers, finally settling on SVM as the most effective classification method [1, 2, 26, 30, 47]. SVM's solid mathematical foundation, further detailed in Annex 1, make it robust against overtraining and the curse of dimensionality. Later papers examined various ways to combine SVM classifiers in order to best perform concept recognition. SVM classification was either performed on a large, multimodal feature vector, in a process called early fusion, or used to combine the outputs of several unimodal classifiers, in a process called late fusion [1, 26, 55]. A final paper describes ontology assisted classification. SVM classification performance is improved by considering semantically related concepts [62]. This theory constitutes the basis for two of the techniques developed in this study, which are presented in chapter 4.

A table on the following page provides an overview of the papers discussed in this chapter. They are categorized by whether the techniques presented are domain specific or generic, the unimodal classifiers used, the meta-classifiers used, the best overall meta-classifier, and the year in which the paper was published.

Table 1 Classification overview

Author	Domain/ Generic	Unimodal Classifiers	Meta-Classifiers Compared	Best Meta-Classifier	Year
[7]	Domain specific (sports)	Feature Based	Rule Based	Rule Based	2003
[20]	Domain specific (animal hunt)	Neural Network	Rule Based	Rule Based	2000
[64]	Domain specific (sports)	Feature based/HMM	Rule Based/ Probabilistic/ Bayesian Inference	Rule Based	2006
[47]	Generic	Feature based	C4.5 decision trees/MaxEnt/SVM	SVM	2005
[2]	Generic	Feature based	HMM	HMM	2001
[30]	Domain specific (hardware package)	Feature based	Combination of individual classifiers vs. SVM fusion	SVM	2002
[1]	Generic	SVM/ GMM/ HMM	Rule Based vs. BN, BN vs. SVM and individual classifiers	BN and SVM outperformed individual classifiers	2003
[26]	Generic	SVM/ GMM/ HMM	BN vs. SVM	SVM	2003
[55]	Generic	SVM	Early fusion vs. normalized ensemble fusion (late fusion with soft decision combinatory logic)	Normalized ensemble fusion	2003
[62]	Generic	SVM	SVM vs. SVM + ontology boosting	SVM + ontology boosting	2004

Chapter 3

Methodology

The initial goal of this thesis was to develop a detector capable of recognizing an abstract high-level concept such as ‘happiness’. This chapter details the basic research that was performed towards that end. This involved choosing a dataset, a supervised learner, and a video annotation tool. The lessons learned from these investigations led to a revision of the initial research goal.

3.1 Basic terminology

The terminology further used is defined as follows. A low-level feature is a piece of audio, video, or text data that has been extracted from a video fragment. Possible examples are color histograms, Mel-ceptstrum coefficients, or a word frequency count. A supervised learner, or classifier, learns to recognize these features and to associate them with a particular semantic concept. A semantic concept is the generic term encapsulating a particular notion or idea. For example, a ‘car’ would be a semantic concept, as it conveys the notion of a particular type of motorized vehicle. In TRECVID terminology, a semantic concept is called a ‘high-level feature’, but that usage is not employed in this paper. Most concepts have a direct link to the feature space. A concept is termed, high-level, when the concept has a particularly abstract definition. A high-level concept cannot easily be recognized by a classifier operating on the existing feature space, although a human may easily be able to do so. This is known as the semantic gap. Concepts such as ‘love’, ‘happy’, ‘sad’, or ‘anger’ are all examples of high-level concepts. Since high-level concepts are not readily detectable from the feature space, than can only be inferred from other concepts. One might even coin the term, intermediate-level concepts, for the concepts that serve as indicators of a high-level concept. For example, ‘crying’ or ‘funeral’ are intermediate-level concepts indicative of ‘sadness’.

3.2 Choosing a video annotation tool

Given a video source, every defined concept requires an associated ground-truth file. This file lists the frames of the video in which a concept occurs, and is required for the machine learning process. The features in the specified frames are used to train a detector to recognize that particular concept. This means, that at a minimum, ground-truth annotations have to be created for the high-level concept that is the goal of this research. Creating ground-truth annotations is a time intensive task, as one must examine each frame one at a time, marking the presence of the desired concept.

The freely available data annotation tool, VideoAnnex [56], was assessed for its potential usefulness in future annotation tasks. It performs annotations on a shot level, which has two benefits. The annotation effort is accelerated, as all frames within a shot share the same ground-truth label. Furthermore, it allows for easier labeling of temporal concepts, that is, concepts whose meanings become apparent over the course of successive frames. VideoAnnex permits region-level annotation, where the user draws a bounding box around a particular area, representative of a concept. Also worth mentioning is that this annotation tool also performs audio playback and therefore allows annotation of concepts which have distinct audio cues. Even with such a comprehensive and efficient tool as VideoAnnex at our disposal, annotation efforts are very time consuming and require a significant investment in man-hours.

3.3 Choosing a dataset

Choosing a video dataset on which to perform experiments is not a trivial issue, as there are a number of factors influencing the decision process. The more abstract the desired high-level concept, the harder it is to create a detector for it, as less low-level features have a direct bearing on the concept. This means that most of the contribution must come from the detection of intermediate concepts, rather than from feature space. For example, the abstract concept 'sadness' might only be inferred from concepts such as 'crying people' or 'funeral'. The immediate consequence of this is that one must also consider whether these intermediate concepts are also present in any dataset. These too then, must have ground-truth annotations created for them. One would need a large digitized video collection to even contain sufficient instances of all the necessary intermediate and high-level concepts, and additionally one would have to make the annotation effort.

This led to consider the TRECVID 2005 corpus, which seemed sufficiently large at 169 hours of news video footage, and had several collections of concept lexicons with associated ground truth annotations. These are: the LSCOM-lite set [35], the MediaMill Challenge set [43], and the complete LSCOM set [32].

LSCOM-lite

The LSCOM-lite set was the result of a common annotation effort by the TRECVID-2005 participants, and contains the ground truth annotations for a collection of 39 concepts. The aim of the LSCOM-lite set was to maximally partition the semantic space, using a minimal amount of concepts, analogous to partitioning the space into a set of hyper cubes. After considering a study of what events were considered newsworthy, the LSCOM-lite developers chose 7 dimensions, each segmented by concepts chosen for their ease of detection and the frequency in which they appeared in search tasks. Most of the concepts from the TRECVID 2003 feature extraction task were included in this set. The annotation software used for this set operated on a static key frame level, thus restricting the concepts to ones that could be identified visually. Temporal concepts, or concepts relying on audio features, could not be used. [35] The deliberate choice for semantically diverse concepts, and the lack of sufficient intermediate level concepts, makes this collection a poor basis for the development of a high-level concept detector.

MediaMill

The MediaMill challenge set augmented the LSCOM-lite lexicon, to arrive at a total of 101 concepts. However, the MediaMill developers maintained the same requirement of visual-only concepts as the LSCOM-lite developers did [35]. Worthy of mention however, is that the MediaMill Challenge set also includes the low-level features with the ground truth annotation of each frame. In addition, optimized detectors are provided for each concept [43].

LSCOM

The LSCOM annotation set, first used for TRECVID 2006, has a larger set of concept annotations, for a total of 856 concepts. However, only half of these actually occur in the TRECVID 2005 video footage. Nonetheless, some of the concepts include in this set are intermediate level concepts, and as such are of greater value towards developing a high-level concept detector. [32]

The full set of LSCOM ground truth annotations offers the best concept lexicon for developing a high-level concept detector. The set is large, and the concepts can be arranged in a

hierarchy that could ultimately be used to deduce the presence of high-level concept such as ‘happiness’, ‘anger’, and ‘sadness’. However, no low-level features or concept detectors were included, and therefore the choice of dataset fell to the more limited MediaMill challenge set.

3.4 Choosing a Supervised learner

The literature survey from chapter 2 lists a number of classification methods that have been used in research to perform concept recognition. They included knowledge-based schemes, Bayesian classifiers, neural network classifiers, Hidden Markov Models (HMM), and Support Vector Machines (SVM).

Knowledge-based approaches in literature were always restricted to a fixed domain, and were not readily extensible to include new concepts. As a result, I rejected this approach, as it seemed unlikely that any rule-based system would perform robustly when tested against a generic video stream.

Comparative studies from the literature survey of chapter 2 have shown that SVM outperformed the above-mentioned methods in terms of classification performance. The success of SVM performance is due to its sophisticated training procedure, which involves mapping input vectors to a higher dimensional space, thus simplifying the task of finding a maximally separating decision boundary. For more specifics, see Appendix A. Besides superior classification performance, SVMs have also been reported as being capable of handling high-dimensional feature vectors without any detrimental effects, as well as being capable of functioning when given only few training examples. For these reasons the decision was made to use SVMs as the supervised learner of choice for the classification experiments performed in this study.

3.5 SVM in practice

Section 3.3 discusses three lexicons of semantic concepts and their associated ground truth annotations. Although the full LSCOM annotation set formed the best basis for defining high-level semantic concepts as it had the richest concept set, only the MediaMill Challenge set was ready for immediate SVM classification experiments given its inclusion of low-level features for each semantic concept in the set. Thus the MediaMill data set is used for the first experiments with SVM classifiers, as the data set permitted the conduct of early and late fusion classification experiments for comparison against the baseline results.

From a literature survey, it transpires that by far the most predominant SVM classifier in use is LIBSVM [12]. The second most cited SVM classifier is SVM-Lite [28], which is optimized and extended with a graphical user interface called SVM-Dark [37]. Both were installed on a 3.2 Ghz home computer. An initial experiment was conducted using SVM-Dark on the MediaMill experiment 3 ‘beach’ concept, which consists of 120 features. SVM-Dark was tasked with finding the optimum parameters for a new ‘beach’ detector. To perform 10 iterations on a reduced instance of the training set took 3 hours. The full 40 megabyte training set ran for over 15 hours, occupied 4 gigabytes of temporary space, and failed to terminate.

A working ‘beach’ SVM classifier was eventually created and run against the provided test set. The results were surprising, so a new ‘beach’ detector was created using LIBSVM, with similar results. Although this classifier had an accuracy of 99.9381% on the test set, all the results were classified as being in the same class. Subsequent SVM classifiers created using both programs for ‘dog’, from MediaMill experiment 4 features, encountered the same problem. The classifiers were scoring highly in terms of classification accuracy, but only placing the test inputs

into one class. Both ‘beach’ and ‘dog’ have very few positive training examples, on the order of <50 while there are over 10,000 negative examples.

The lack of positive examples means it is very difficult to train SVM classifiers sufficiently capable of recognizing the ‘beach’ and ‘dog’ concepts. The challenge is in discovering the optimum parameters for the classifiers. (See Appendix A for further information about parameters that influence the creation of a SVM classifier.)

The use of SVM-Dark was discontinued, as LIBSVM was better suited at finding the optimum classifier parameters. On average, parameter learning took between 4 to 10 hours per concept. Eventually successful classifiers for both ‘beach’ and ‘dog’ were created after an exhaustive search for the correct kernel parameters.

3.6 Adjustment of the research approach

The use of any publicly available video source was precluded by the lack of ground truth annotations, and annotating a video source by hand would have proved to be too labor intensive. This led to the examination of three collections of ground truth annotations of the TRECVID 2005 corpus. Of these, the MediaMill dataset was chosen because it was the only collection to contain both the ground truth annotations and the features of each frame, as well as optimized detectors for each concept. Although it would have been possible to create detectors for the concepts in the full LSCOM collection given the MediaMill features, this would have been too computationally intensive. The MediaMill concept lexicon, however, was more limited than the LSCOM collection, and did not have many concepts which semantically indicated a high-level concept. Cursory experiments with the SVM classifier had shown that it was quite hard, and time consuming, to get decent detection results for even some simple concepts within the MediaMill dataset.

The lack of concepts semantically indicative of a high-level concept, and the poor detection results of these basic concepts suggested it was unlikely that any combination of simple classifiers could feasibly be used to create a high-level concept detector. This led to a shift from the original research goal. Instead of creating a high-level concept detector by detecting and relating the underlying concepts, the aim of the study would be to improve existing concept detectors by considering the presence of semantically related concepts.

Chapter 4

Inter-conceptual boosting experiments

There are various ways to improve the performance of a generic SVM concept detector. Simply selecting better training parameters when generating the SVM model will yield an improvement. Other possibilities are applying different classification schemes such as early or late fusion. The following three experiments aim to improve detector performance by use information about the relationships between concepts. Semantic relationships, as modeled in an ontology, are used by the Ancestor boosting and Sibling-confusion removal techniques. Concept correlations garnered by a Chi-square test, are used by the Chi-square boosting technique. These techniques are used to develop new detectors for each concept in a dataset. These new detectors are compared against the original detectors for each concept, to see whether there is an improvement in the mean average precision (MAP) scores. These scores will be reported and analyzed in order to better understand the effectiveness, and shortcomings, of each technique.

4.1 Experiment Setup

The subsequent experiments were performed on the MediaMill dataset, using the 120 features and detectors from the MediaMill Experiment 3 collection. This particular collection was chosen because it used all possible features (Experiment 1 examines graphical features only, Experiment 2 examines textual features only) and thus the link between features and semantic content in each shot seemed most complete and least indirect (Experiment 4 combines the scores from Experiments 1 and 2, adding a layer of indirection). Early fusion detectors were used as a baseline detector for each concept. Training was performed on a set that consisted of 70% of the data, and results were computed against a test set, consisting of the remaining 30% of the data [43].

The mean average precision (MAP) score is used to compare various concept detector results. This value is computed by taking the average of the precision scores of the relevant shots from a ranked list of detector confidence scores.

Let Precision(i) be the precision at rank i, where precision is defined as the number of relevant and found shots over the set of found shots. Let Relevant(i) be a function which states whether the shot at i is relevant. Then for a concept with N shots of which #relevant are relevant, the MAP score is defined as:

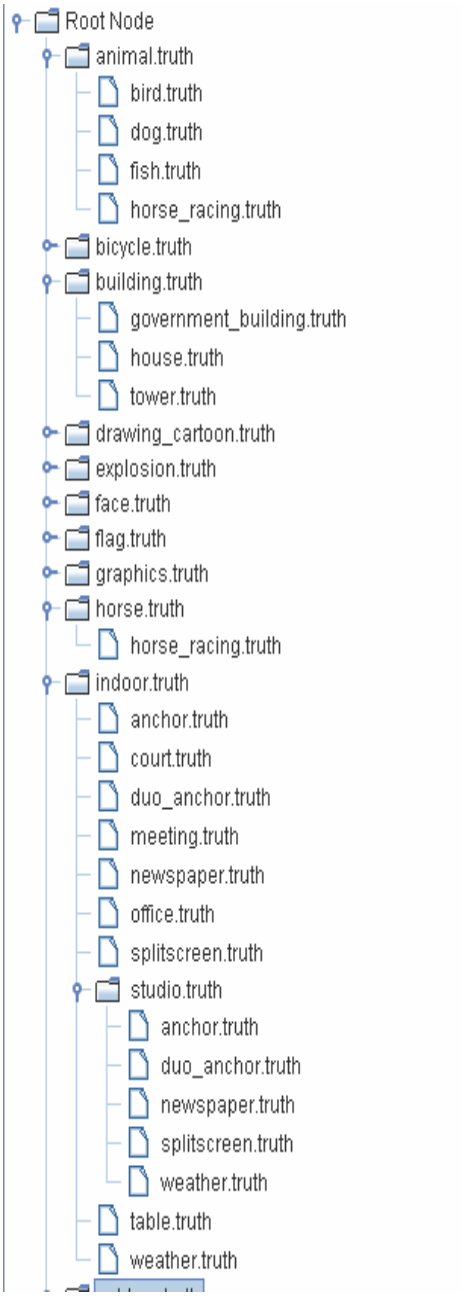
MAP: $1/\#\text{relevant} * {}_{i=1}\sum^{N} (\text{Precision}(i)*\text{Relevant}(i))$

Mean average precision is a useful metric as it combines precision and recall into one single value. MAP emphasizes returning more relevant shots earlier, and as such is an appropriate choice of metric for comparing concept detectors.

A dictionary defines words in terms of other related words. Similarly, the presence of one concept could indicate the presence of a related concept. The following experiments detail three unique approaches to modeling inter-conceptual relationships to boost individual classifier performance. The first two experiments place the concepts present in the data set into a tree hierarchy, based on a statistical analysis of their occurrences, resulting in an ontology. This is a structured approach to modeling the relationships between various concepts, akin to a dictionary in real life. Once in a tree structure, the concepts on the sibling and ancestor-child axis' are consulted when generating a concept classifier. The third experiment generates an unstructured set of highly correlated concept pairs present in the data set. When generating a concept detector,

the presence of a highly correlated concept can be used to distinguish ambiguous low-level features.

The first step is to generate the ontology itself, as in the adjacent figure. This was done by calculating the posterior probabilities of each concept against the others in order to determine which concepts were supersets of the others. Given a posterior probability P(A|B), concept A was placed as an ancestor node in the ontology and B as a child node of A when the posterior probability exceeded a certain threshold value. For this experiment, the threshold was set at 95%. This results in a natural hierarchy that reflects the relationships of the concepts within the data set. The focus of the following experiment was to enhance concept classifiers which shared the same sibling axis. Examples of this are the concepts {government, building, house, and tower}, which have the 'building' parent concept. The concepts are related semantically on the parent axis, but are semantically mutually exclusive on the sibling axis. Since only posterior probabilities were used to generate the ontology, this procedure is inadequate to definitely conclude that sibling classes are mutually exclusive.



4.2 Experiment 1: Sibling-confusion removal

Sibling concepts are semantically very related, and this tends to be reflected in their feature sets, which also are likely to be very similar. This causes confusion among their concept detectors, which are unable to distinguish the features correctly, resulting in many false positives. Based on work by Wu [48], the Sibling-confusion removal technique reduces the amount of false positives detected by normalizing detector scores based on the confusion factor, a number that indicates the likelihood of a false positive occurring for a particular shot.

This experiment assumes once all the concepts have been placed in the tree-hierarchy and the ancestor-child relationships have been determined, that the set of sibling concepts of a sub-tree is distinct and complete. The assumption of mutual exclusivity of sibling concepts is crucial to the experiment. If a shot has been classified as a member of the parent class e.g. {building}, then it must be one of the specific child classes, either: {government building, house, tower}. However, as the specificity of the concept increases, so does the scarcity of positive training examples resulting in less robust concept detectors. As such there is a significant chance that shots may score highly among several sibling concept detectors, especially if there is little to distinguish between the concepts, resulting in several false positives. This is known as the confusion factor [48]. This experiment focuses on modifying sibling concept detectors to deal with the confusion factor. By reducing the amount of false positives between sibling concept detectors, the mean average precision of each detector will be increased.

Intuitively, the confusion factor is only an important consideration when a shot scores highly on two or more sibling class detectors. Only one detector can be correct, and the others must be detecting false positives (because mutual exclusivity is assumed on sibling classes in the ontology, due to their semantic meanings). Formally, given shot s, concept C_i, and the set of concepts $C_{sibling}$, which are the sibling concepts of C_i, the confusion factor is defined as $P(s|C_i)$-max($P(s|C_{sibling})$), the difference between the probability that s is an example of C_i and the highest scoring sibling concept. $P(s|C_i)$ is the confidence score given by the SVM classifier for that concept. For this experiment, an early fusion (MediaMill experiment 3) type classifier was used, that learned each concept from 101 low-level features.

The initial confidence score $P(s|C_i)$ for each concept in the data set must be updated to take the confusion factor into account. As such, the general equation for this update becomes:

$$P(s|C_i)_{updated} = P(s|C_i)_{initial} * f(\text{confusion factor}),$$

where the confidence score is updated as a function of the confusion factor:

$$P(s|C_i)_{updated} = P(s|C_i)_{initial} * |f(P(s|C_i)_{initial} - \max(P(s|C_{sibling}))|$$

The function max simply returns the highest scoring sibling concept, $P(s|C_{sibling})$), of C_i.

Several considerations, such as the magnitude and sign of the confusion factor, have to be taken into account when determining the function. A small confusion factor implies that there are two highly scoring concept detectors (or trivially two low scoring ones). This in turn means that $P(s|C_i)_{updated}$ ought to be significantly smaller, as to reflect the uncertainty of placing shot s in the correct concept class. A large, positive confusion factor on the other hand, implies that shot s is most likely of class C_i and not likely to occur in any of the sibling classes. As such, $P(s|C_i)_{updated}$ must remain large. The large, negative confusion factor case implies that shot s is most likely a member of $C_{sibling}$ and unlikely to be a member of C_i. As such, it is safe to reduce the score of $P(s|C_i)_{updated}$ even more. Finally, one must consider the rate of change between each of the extremes presented above. A large positive confusion factor requires that $P(s|C_i)_{updated}$ stay large, while a small confusion factor requires a sharp drop in $P(s|C_i)_{updated}$. This suggests that the relationship between confusion factor and $P(s|C_i)_{updated}$ is non-linear, and in fact ought to be exponential. The function $f(x)=e^x$ meets all these requirements resulting in:

$$P(s|C_i)_{updated} = P(s|C_i)_{initial} * e^{(P(s|Ci)initial - max(P(s|Csibling))}$$

The final step is to normalize the results back into the range [0-1]

$$P(s|C_i)_{updated} = P(s|C_i)_{initial} * e^{(P(s|Ci)initial - max(P(s|Csibling))} / e$$

In essence, this formula can be thought of the normalization of confidence scores based on the interference from closely related sibling classes. This formula was used to update the confidence scores of the sibling concept detectors in the MediaMill dataset, for which afterwards the MAP was computed. The results obtained by executing Sibling-confusion removal on the test set are presented in Annex 2, with an analysis thereof in the following section, 4.3.

4.3 Sibling-confusion removal analysis

In this experiment, 30 out of 64 concepts had improved MAP scores. All concept groupings but the {indoor, outdoor} set showed a general decrease in overall MAP after the confusion removal technique had been applied. A few concepts had a minute increase, but only in the third or fourth decimal place, so as to be negligible. Other concepts in those groupings show a comparatively larger decrease. Only groupings in which all the concepts showed an improved MAP score were actually considered improved by this technique. This does beg the question whether the confusion removal technique actually works. The increase in the MAP of the concept ‘indoor’, from 0.59 to

0.60, and the concept 'outdoor', from 0.71 to 0.72, should be taken as a qualified success however. What then, however, are the reasons for the general decrease in MAP over many sibling-concept groupings?

At the start of the experiment, it was hypothesized that all sibling groupings had to be mutually exclusive. More correctly, such a set has to be closed, and each concept within has to be complementary to all the others. That is to say, assuming parent concept C_p was detected in shot s, one, and only one, of the child concepts that made up the sibling concept set also had to be detected in the shot,

$\forall C_i \in C_p \bullet C_i$ occurs in s$\rightarrow \forall C_{sibling} \in C_p \bullet C_{sibling} \neq C_i \rightarrow C_{sibling}$ does not occur in s.

Although these properties fortuitously hold for the set {indoor, outdoor}, this however could not always be enforced for other groupings.

In part, the fault lies with the automated ontology determination process. Although it was able to determine parent-child relationships, it was not able to adequately group the various sibling concepts according to their semantic relationships. As a result, 'tony blair' was omitted from the government leader set of {allawi, arrafat, bush_jr, bush_sr, hu_jintao, kerry, lahoud, powell}, thus violating the desired closure property. Likewise an unrelated group of siblings such as {anchor, duo_anchor, newspaper, splitscreen, weather} was entirely possible too, thus violating the property that each element had to be the complement of the remainder of the set. The failure to make the correct sibling sets at the semantic level means that the confusion removal algorithm is doomed to failure prior to execution.

The following example illustrates the effects of violating the closure rule when creating set groupings. In this scenario, the sibling concepts are $C_{i,}$ $C_{j,}$ and $C_{missing}$. $C_{missing}$ is so termed because it is alternately included in the set of sibling concepts in the ontology.

Consider again the formula: $P(s|C_i)_{updated} = P(s|C_i)_{initial} * e^{(P(s|Ci)initial - max(P(s|Csibling))}/e$.

Let shot s contain the concept $C_{missing}$, and let the initial detector scores be $\{P(s|C_i)=0.40, P(s|C_j)=0.41, P(s|C_{missing})=0.9\}_{initial}$.

The first case is when $C_{missing}$ is included in the sibling concept set. Because C_i and C_j are semantically related to $C_{missing}$, they get an ambivalent detector score on shot s. However, because shot s scores so highly on concept $C_{missing}$, there is a sharp drop in their modified predictions after confusion removal, $\{P(s|C_i)=0.089, P(s|C_j)=0.092, P(s|C_{missing})=0.54\}$.

The second case is when $C_{missing}$ is omitted from the sibling concept set, the scores then after confusion removal are: $\{P(s|C_i)=0.145, P(s|C_j)=0.152\}$, which is less of a decrease in confidence estimations. This in turn impacts the mean average precision score, as it computes, per concept, the average precision over a list sorted according to confidence scores. Thus $P(s|C_i)=0.089$ ($C_{missing}$ not omitted) would correctly rank lower on the list than $P(s|C_i)=0.145$ ($C_{missing}$ omitted).

The other case to be considered is when the elements in a set are not all complementary to the rest, per the requirement: $\forall C_i \in C_p \bullet C_i$ occurs in s$\rightarrow \forall C_{sibling} \in C_p \bullet C_{sibling} \neq C_i \rightarrow C_{sibling}$ does not occur in s. Consider three concepts, $\{C_k, C_{l,}$ and $C_m\}$. Concepts C_k and C_m are sibling concepts, while C_l is an independent concept, unrelated to both. Let shot s contain the concepts C_k and C_l, and let the resultant detector scores be $\{P(s|C_k)=0.93, P(s|C_l)=0.92, P(s|C_m)=0.11\}$ initial. C_k and C_m are complementary over shot s, which is reflected in their confidence scores, and approximate 1 and 0 respectively. C_l is independent of either however. Because it is present in the sibling group its confidence score is included in the calculation, which entirely ruins the updated scores and the resultant MAP score, $\{P(s|C_k)=0.34, P(s|C_l)=0.33, P(s|C_m)=0.02\}$ (incorrect- C_l is included) Omitting C_l however, gives the following scores, {P(s|Ck)=0.77, P(s|Cm)=0.02}

(correct- C_l is excluded). In a rank ordered list, the incorrect scores, from the inclusion C_l in the calculations, would be much lower on the list, relatively, and would thus lower the MAP score.

Thus the presence of an extraneous concept or the absence of one as a result of a poorly constructed ontology can lower the MAP scores of all the concepts in the resultant sibling sets. This undoubtedly played a significant role for the generally lower scores, but is not the entire explanation, as even groupings which have been amended to ensure the set properties discussed above perform worse after confusion removal. Examples of this are the sets {male, female} or {drawing, cartoon}

There is a simple, numeric explanation that also must be considered. Confusion removal is a normalization operation, due to the division of the re-ranked score by e.

Recall:

$$P(s|C_i)_{updated} = P(s|C_i)_{initial} * e^{(P(s|Ci)initial - max(P(s|C\ sibling))}/e.$$

Inherently, re-ranked scores are lower than their initial values, due to the exponential, $e^{[0-1]}<e$. The insight however, is that shots which have a lot of confusion, have an exponentially lower value, so in a rank-ordered list, the only positions which change are those of shots with a lot of confusion amongst related concepts. In theory, this should mean that the mean average precision scores for all concerned concepts should stay static, or improve as a result of the lower positions in the ranked list of the false positives. How come then, that the {male, female} set has decreased MAP scores overall? 'Male' has an initial MAP of 0.0678, which drops to 0.0675, and 'female' has an initial MAP of 0.0609, which drops to 0.0405. The simple answer is that sometimes the detectors themselves are not up to the task. In this example, both the male and female detectors gave low confidence estimates, which tended to 0. Shots which contained the concept and shots that did not, received indistinguishable estimates. The lack of detector certainty is compounded in the exponent, which approached 0, resulting in a larger drop in the ranked list.

The final consideration is that the confusion removal algorithm only works if there are, in fact, false positives to remove. If there are none, the algorithm only causes a decrease in the MAPs of the concepts under consideration. An example of this occurrence is the animal grouping, consisting of {bird, dog, fish, horse}. All set properties discussed previously are valid for this grouping. Nonetheless, the MAP of the bird concept drops from 0.761 to 0.744, the MAP of dog stays at 0.103, the MAP of fish stays at 0.407, as does the MAP of horse at 0.0003. The ideal case would be detectors that gave confidence scores: $P(s|C_i)=1$ and $P(s|C_{sibling}) =0$. In practice however, $P(s|C_i)<1$ and $P(s|C_{sibling})>0$ resulting in a non-trivial value in the exponent, $e^{(P(s|Ci)initial - max(P(s|Csibling))}/e$, which causes a change in the position in the ranked list of a shot-score, and ultimately a decrease in the concept MAP.

Ontology assisted confusion removal is a tool that employs the semantic relationships between sibling concepts in order to adjust a detector's confidence level depending on the likelihood of a misclassification. In the current data set, only the {indoor, outdoor} set showed noticeable improvement, while the {beach, river, swimming pool, waterfall} showed a very mild improvement. Nonetheless, confusion removal has potential, although it also has some requirements that make it fragile to deploy, too fragile for this data set at least. Firstly, it requires a concept grouping that is semantically related, closed, and where, for every shot, each concept is complementary to the set of remaining concepts. This is not a major obstacle, but does require human intervention to achieve. Secondly, there is a basic level of performance required by the concept detectors of a sibling set before any positive improvement is noticeable. The effect of weak concept detectors is compounded, to the detriment of all, by this technique. Finally, there must be a certain amount of false positives in the concept set under consideration for the technique to be worthwhile. If the number is insufficient, the MAP might even decrease slightly.

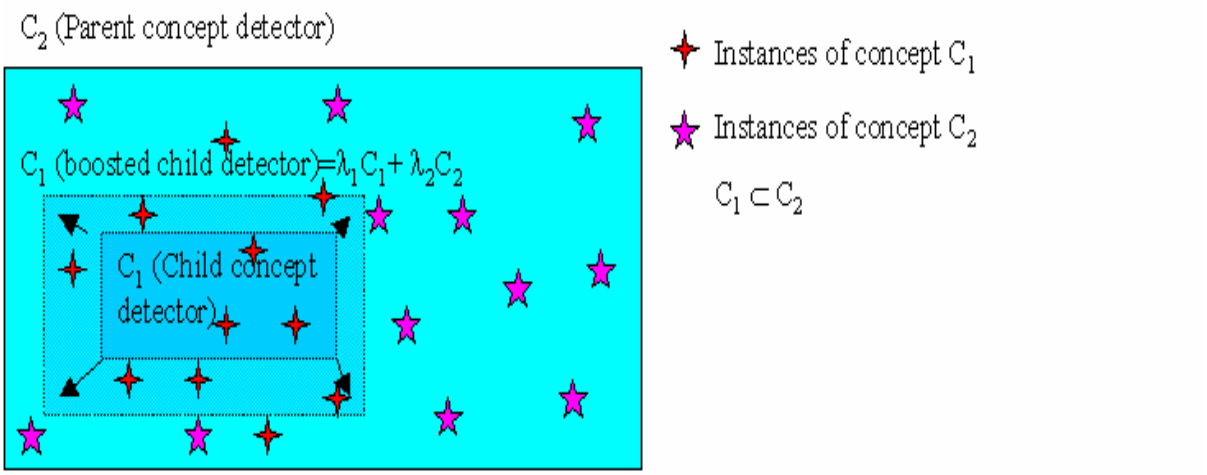
The last point is not too much of an issue, because Sibling-confusion removal runs in linear time with respect to the size of the dataset, and executes quickly. As such, a trial-and-error approach is sufficient to determine the concepts that benefit from the application of this technique.

4.4 Experiment 2: Ancestor boosting

This experiment builds upon the concept ontology from the previous experiment. Instead of considering sibling concepts it focuses on the parent-child relationship between concepts. That is, the concern here is the concepts which are semantically related, where a parent concept is the superset of the child concept. For example, the concept 'water body' encompasses the concepts beach, river, swimming pool, and waterfall in their entirety. The goal is to improve the performance of a child-concept detector by considering the results from the related parent-concept detector. This is a more detailed examination of a technique first done by Wu [48].

Fewer concept examples are present in the dataset in general when a concept is semantically more specific. This adversely impacts the training phase of such a concept detector, and results in a less robust detector. Ancestor concepts, however, have broader semantic definitions and thus occur more frequently in the data set, and as such are detected more robustly. The idea behind this experiment is to leverage the performance of the more accurate and powerful ancestor-concept detectors to boost the performance of their child-concept detectors. Since the ancestor and child concepts are semantically related it seems plausible to combine the results of their detectors. Thus one compensates for a less accurate, but specifically targeted, child concept detector by using a more robust ancestor detector with a broader semantic coverage.

Thus a child concept detector is a linear interpolation of all the ancestor detectors, whose weights are determined in a training phase. The figure below illustrates the idea.



Expressed formally, the updated confidence detector for child concept C_i, given shot s and ancestor concepts $\{C_j \ldots C_k\}$, would have the following formula:

$$P(s|C_i)' = \lambda_i P(s|C_i) + \lambda_j P(s|C_j) + \ldots + \lambda_k P(s|C_k)$$

The parameters λ are determined by empirically finding the optimum weights through the use of the Expectation Maximization (EM) algorithm detailed in [34] and presented below:

1. A concept grouping is selected, consisting of a child concept $\{C_1\}$, and its ancestors, $\{C_2 \ldots C_n\}$.

2. Each $\lambda_1 \ldots \lambda_n$ is initialized with a random number in the range [0-1].
   Steps 3 and 4 are iterated until λ_i converges:
3. For every concept $C_i \in \{C_1, C_2 \ldots C_n\}$
   For every shot s in the set of shots predicting the child concept, C_1
   $\beta_i = \sum_{s \in S}(\lambda_i P(s|C_i) / \sum_m \lambda_j {}^m P(s|C_j)^m)$
4. $\lambda_i = \beta_i / \sum \beta_j$

After sufficient iterations, λ_i will have been determined. These can then be used to update the shot scores for that particular child concept detector.

New child concept detectors were created for all child concepts in the MediaMill dataset by interpolating various ancestor-child concept combinations. The λ parameters were determined for all concepts in a particular ancestor-child set, by running the EM algorithm detailed above against the training set. In this experiment, 30 iterations were done. MAP scores for each concept were determined after performing Ancestor boosting on them, using the recently discovered λ parameters on each ancestor-child concept in the test set. The results are presented in Annex 3 and analyzed in 4.5. The original MAP scores for child concepts are listed, followed by the MAP scores after ancestor boosting. Some concepts in the training set did not have scores in the full range [0-1], and were first normalized for compliance before the EM algorithm was applied. This affected the λ values, and changed the resultant MAP scores, which is also displayed. For comparison purposes, sometimes an ancestor concept was omitted from a sub tree.

4.5 Ancestor boosting analysis

In this experiment, 17 out of 61 distinct concepts had improved MAP scores. The assumption behind this experiment was that child concepts only occur rarely in a dataset, and thus have weak concept detectors. These child detectors are insufficiently able to recognize the low-level features that determine the semantic meaning of the concept. Ancestor concepts, on the other hand, occur more often, and thus have stronger detectors. Since semantically a child concept is automatically a subset of the parent concept, the assumption is that the low-level features that determine the child concept are also the features that determine the parent concept. Thus it should be possible to interpolate the scores from the ancestor and child concept detectors for a particular shot, and arrive at a more definitive detector than the original. This should be conceived of as the expansion of the classification boundary of the child detector, based on the classification boundaries of the ancestor concepts. This section analyzes and reflects upon the results garnered from the application of the ancestor boosting method on the MediaMill dataset.

Of the 94 sub-trees, 18 concepts had better MAP scores when normalization occurred before boosting, 11 had worse scores, and the remainder were unaffected by normalization. Since the MAP results with normalization were generally slightly better, these values are cited in subsequent passages. Out of 94 sub-trees, 21 gave improved results after ancestor boosting, while the remainder performed worse. The most noticeable improvements were the sub-trees {desert, outdoor}, where the 'desert' MAP improved from 0.093 to 0.183 (96% increase), {anchor, studio, indoor}, where the 'anchor' MAP improved from 0.619 to 0.635 (2.6% increase), and {swimming pool, water body}, where the 'swimming pool' MAP improved from 0.0014 to 0.0054 (285% increase). The remaining 73 sub trees, whose detectors performed worse after boosting, had MAP scores that sometimes varied mildly, but sometimes by as much as one significant figure.

There are two possible causes for the variations in MAP amongst the sub-trees:

- The semantic distance between concepts; and
- Stronger child detectors than ancestor detectors.

The semantic distance is the subjective measure of the closeness in meaning of two concepts. In the context of the ontology of the dataset, ‘building’ is semantically close to ‘house’, while ‘building’ is rather more semantically distant from ‘outdoor’. Although these concepts share the same ancestor-descendant axis, they differ considerably in meaning. A ‘house’ is an instance of a ‘building’, while a ‘building’ is located ‘outdoors’. Thus the first two are very similar, while the latter is a much more generic location. As such, the probability of the ‘house’ and ‘building’ concepts co-occurring in a shot is very high. The likelihood that ‘outdoors’ and ‘building’ co-occur is considerably less.

This has bearing on the interpolation process, where the various child and ancestor detectors are interpolated together to form a new detector. The EM algorithm determines the optimum weightings for the contribution from the child and ancestor detectors. If some of these ancestor-child concepts are not concurrent then the incorrect weighting is found, resulting in an overall MAP decrease for the interpolated detector.

This can be seen in the results, as the ‘house’ decreases when boosted by its ancestors ‘building’ and ‘outdoor’, from 0.00664 to 0.00605, but increases to 0.00804 when boosted by ‘building’ alone. Likewise, the MAP for the ‘anchor’ detector improves when boosted by ‘studio’, from 0.06192 to 0.6358, but only improves to 0.6354 when boosted by both ‘studio’ and ‘indoor’. It can be argued that an anchor is much more closely related to a studio, than to the location ‘indoors’. Thus semantically close ancestors boost a child concept, while semantically remote ancestors only detract from it.

The set {basketball, walking and running, people} is an interesting examination of the issue of semantic distance. These concepts seem related, and one would suspect an increase after ancestor boosting. The resultant drop in MAP, from 0.1791 to 0.0054 suggests otherwise. For this sub-tree, parameter estimation greatly favors the ‘people’ concept, and the resultant detector is more of a reduced ‘people’ detector than one that can recognize ‘basketball’. In reality, the semantic distance, as a measure of concurrency, between ‘basketball’ and ‘people’ is considerable. In the dataset, most drops in MAP performance are caused by the semantic dissimilarity between ancestor and child concepts.

Some however, are simply caused by over-interpolation. The premise of ancestor boosting is that the child-concept detector is insufficiently trained, and that the semantically similar ancestor detector is more than up to the task. In that case, linear interpolation is entirely appropriate. However, for some concepts in the dataset, the child detector is entirely capable, and the ancestor classifier is the weaker classifier. The more abstract ‘water body’ and ‘animal’ concepts are examples of this. Interpolating with these concepts only cause a loss in specificity in the resultant detector, as is illustrated in the drops in performance for the {river, water body} detector, from MAP 0.653 to 0.253, and the {bird, animal} detector, from MAP 0.761 to 0.747.

Ancestor boosting is a promising technique, as evinced by the number of improved child detectors in the dataset. For ancestor boosting to work successfully, however, it is necessary that ancestor and child concepts be tightly linked semantically, as this implies a high degree of concurrency. Some concepts in the dataset contradicted the notion that ancestor concepts would have better performing detectors than their child concepts. Where this was the case, the resultant interpolated detectors performed worse than the original concept detector. An interesting research question would be to see how much the performance of various child concepts would improve were the dataset seeded with additional intermediate level ancestor concepts. Another research idea would be to perform the ancestor boosting with a SVM classifier, instead of linearly interpolating the detectors. That is, the SVM classifier would take the outputs of the child and ancestor classifiers, and internally generate a combinatory classifier. There is a risk that the EM

algorithm for the λ parameter training over-trains, and that a SVM classifier would be more robust under different testing conditions.

4.6 Experiment 3: Chi-Square boosting

In the previous two experiments, an ontology was used to determine the relationships between concepts, prior to concept boosting. Inspired by Yan [67] and Hauptmann[23], who used the chi-square test to find related concept pairs, this experiment determines concept correlations through the application of this method. The chi-square test does not imply a structured relationship, as the tree-hierarchy of an ontology might, but simply identifies concept pairs that are significantly related, in other words, concepts that frequently occur together in the same shot.

The assumption made is that the training set, containing the ground-truth values on which the chi-test operates, is representative of the larger dataset. Provided this is true, the concept relationships learned from the training set can be used to improve the detectors of the whole set.

The insight behind this method is that if a concept is significantly related to another concept, then the chance of the other related concept also appearing in a shot is very high. In essence, one could use the detector for one concept and still detect the related concept, simply because the presence of one implies the other. Thus one could compensate for a poorly performing concept detector by using the detector from the related concept. If the representative features of each of the concomitant concepts are fairly independent of each other, a combined detector could potentially perform more robustly for both concepts because it would perceive both sets of representative features. If one set of features was feint, it could still detect the other set of features.

4.7 Chi-Square explained

Pearson's chi-square test was used to evaluate each concept pair in order to determine whether they had similar frequency distributions for their ground-truth values. The chi-square is the result of the sum of all the squares of the difference of the observed frequency and the expected frequency, divided by the expected frequency. This is expressed in the following formula:

$\chi^2 = \sum_{i=1}^{n} (O_i - E_i)^2 / E_i$ where O_i is the observation, and E_i the expected value, at the i-th element in a table with n elements.

In order to get meaningful results this entailed partitioning the ground truth set, and for each concept-pair, drawing up contingency tables that detailed the number of concept occurrences in each partition. For example:

#of observed occurrences	Frames 1...k	k+1...2k	…	n-k+1…n	
Concept 1	2	4	…	6	Row total=12
Concept 2	1	2	…	3	Row total=6
Totals	Column total=3	Column total=6	…	Column total=9	Grand total=18

Thus for the above example, where concepts 1 and 2 are subdivided into partitions of size k:

Let k=3

The expected value is computed by taking the row total(i)/grand total*column total(i). E.g. for i=4 (the above table has 6 elements in total-the 4th element is the first column of the second row) $E_4=(6/18*3)$

$\chi^2=(2-(12/18*3))^2/(12/18*3)+(4-(12/18*6))^2/(12/18*6)+\ldots+(6-(12/18*9)^2/(12/18*9)+$
$(1-(6/18*3))^2/(6/18*3)+(2-(6/18*6))^2/(6/18*6)+\ldots+(3-(6/18*9)^2/(6/18*9)=0$

If χ^2 is less than a threshold value, determined by consulting the chi-square distribution for the degrees of freedom and the significance value, then the two concepts are deemed related. The significance value used for the chi-square tests in this experiment was 0.05, as this is considered the criterion for statistical significance. There are (k-1) degrees of freedom in the above example. Consultation of the chi-square distribution for (k-1) degrees of freedom and a significance value of 0.05 gives a value of approximately 5.991. Because $\chi^2=0<5.991$, the two concepts are related. In practice, the row totals are determined from the ground-truth annotations for each concept, and the column totals are calculated at run time, as is the degree of freedom.

In order to perform a meaningful chi-square test, the key-frames and their ground-truth values had to be partitioned into smaller sets. The partition size for these sets had to be carefully chosen. There are concepts in the MediaMill dataset with fewer than 100 instances so the partition size had to be significantly less than that. One criterion for the validity of the chi-square test is that each cell has to have at least 5 observations. A decrease in the partition size results in more concept-pairs being identified, at the expense of an increase in the number of false positives as the 5-observation requirement is violated. This is especially likely with sparse concepts. The following example illustrates the effect of a decrease in partition size. Consider two concepts, with a partition size of 2 on their ground-truth values:

Concept 1	TT	FF	TT	…	FF
Concept 2	TT	FF	TF	…	FF

For the sake of the argument, say that these two concepts are not related according to the chi-test, because even though the first column has an equal amount of true shots for both concepts, they differ in the third column. Partitions containing only false shots are ignored. With only one out of two partitions in common, the frequency distributions of the two concepts are too dissimilar to consider them significantly related. Decreasing the partition size from 2 to 1 would cause the following:

Concept 1	T	T	F	F	T	T	…	F	F
Concept 2	T	T	F	F	T	F	…	F	F

Now the two concepts have three out four partitions in common, which would be cause enough to consider them significantly related. A small enough partition induces a similar frequency distribution between concepts, because of the trivial amount of elements per partition. Hence the 5-observation requirement as the minimum size for a partition in a chi-square test. For video retrieval, it should probably be larger than that.

With a partition size of 10, 55 concept pairs were identified, although a number of these were false positives, i.e. {duo_anchor, clinton}. Instead, a partition size of 40 was used for the experiment, which identified 23 concept pairs.

New concept detectors were created for each of the reported concepts, by interpolating the concepts of each pair using the procedure first reported in section 4.3. When a specific concept appeared in more than one pair, a detector was created by interpolating all the related concepts. The MAP results for the updated concept detectors are presented in Annex 4. Concept pair generation was done using the whole data set, i.e. both the training and test sets combined, in

order to get a fairer idea of concept relationships. Parameter training, as per 4.3, was done on the training set alone though.

4.8 Chi-Square boosting analysis

The 23 identified concept pairs were made up of 36 unique concepts. 22 of these 36 concepts had detectors who showed an improvement in MAP. 46 detectors were created as a result of pair-wise concept interpolations, and 9 concept detectors resulted from interpolations of more than two concepts. The MAP results obtained with a normalized training set were better than those without normalization, and so the normalized values are used for this analysis.

27 of the 46 concept pairs showed an improvement after interpolation. Of the 27, 18 concepts in a pair both improved. It is interesting to examine a few concept pairs, and to speculate about their results.

For the sets: {horse, horse_racing} and {flag, flag_usa}, 'flag_usa' and 'horse_racing' are in fact strict subsets of 'flag' and 'horse'. In fact, this had already been established in the ontology and ancestor boosting section of 4.2. More interesting is that the ancestor detectors perform better when combined with their child detectors. This is contrary to the assumption that ancestor detectors are stronger than their child detectors, which was made in the Ancestor boosting section. A possible explanation for this behavior may rest with the distinguishing features for the semantic content of each concept, which varies slightly, and the number of training examples for each concept. 'Horse_racing' is a 'horse' detector, with an added motion component. Likewise 'flag_usa' is a 'flag' detector, with specialized color components specific to the American flag. Child detectors that detect specialized feature components with a high degree of certainty boost the ancestor concept detectors, 'horse' and 'flag'. The child components, 'horse_racing' and 'flag_usa', benefit from ancestor detectors more capable of detecting the generic concept. In essence, both ancestor and child concept detectors engage in a sort of mutual error compensation. A possible explanation for why these detectors are able to engage in mutual error compensation may be that the individual detectors are not expert enough, due to a lack in positive training examples. In the whole set, there are around 500 positive examples for 'flag', 400 for 'flag_usa', around 50 for 'horse' and 40 for 'flag'.

Different cases are concept pairs: {building, tower}, {building, house}, {graphics, charts}, and {graphics, maps}. Although 'building' and 'graphics' are still the supersets of {house, tower} and {charts, maps} respectively, they are larger than the combined sum of these child subsets. The child concepts improve because of the contribution from more powerful ancestor detectors, per the reasoning originally discussed in the Ancestor boosting section. More interesting is the improvement of the superset concepts, 'building' and 'graphics'. Using 'building' as an example, one should realize that 'house' and 'tower' make up a large proportion of the 'building' set, with a minor feature contribution from other concepts. The generic 'building' detector thus is capable of recognizing the various specialized building instances. The 'building' detector becomes much more specialized, however, after interpolation with either 'house' or 'tower'. Since the detector is better able to recognize the specialized building instances that occur frequently, i.e. it is less confused by the features from the infrequently occurring sub-types; there is an overall increase in detection performance. This is illustrated by the fact that the MAP increase for the interpolated detector resulting from {house, tower} is greater than the increase for {house, building}, and that there are many more instances of 'tower' in the 'building' set, than there are instances of 'house'. Of note is that the interpolated building detector {building, house, tower} specializing in recognizing the most predominant building subtypes, 'house' and 'tower' gives a MAP of 0.23409, thus outperforming the specialized detectors resulting from interpolating {building, tower} or {building, house} alone. The 'graphics' concept behaves similarly. A MAP gain, from

MAP: 0.38149 to MAP: 0.38153, is seen for ‘graphics’ when specializing it on ‘maps’, which is the most predominantly occurring subtype. A MAP decrease is observed when specializing it on the less frequently occurring child class, ‘charts’. Also specializing ‘graphics’ on both ‘charts’ and ‘maps’ leads to a MAP decrease.

The final category of observations pertain the concept pairs which are disjoint, such as {lahoud, chair}, {military, fire weapon} or {duo_anchor, swimming pool}. Neither concept is a subset or superset of the other, and has only been identified because the chi-square test has noticed their similar frequency distributions. They did not appear in the ontology used in the Sibling-confusion removal and Ancestor boosting techniques, nor would one make the association giving the semantic connotations of each concept. These pairs simply occur because many shots which contain ‘lahoud’ also contain a ‘chair’. The fact that both the revised detectors for ‘chair’ and ‘lahoud’ show improvement suggest there is merit in detecting for disjoint concepts that often occur together. Like the {flag, flag_usa} concept detectors, the revised detectors for ‘chair’ and ‘lahoud’ are more robust as they detect for both sets of representative features. However, the same sources of error for {flag, flag_usa} are also possible for {lahoud, chair}, as all concepts have relatively few positive training examples. An additional potential source of error is the size of the partition. A different partition size might result in a chi-square test that does not consider {lahoud, chair} to have similar frequency distributions. Still, the improvement in MAP mitigates the argument.

One can make a number of categorizations about the types of MAP improvements observed in the results.

- Child (subset) concepts leveraged the functionality of their more powerful ancestor concept detectors. These concepts are identical to that of Ancestor Boosting, section 4.3, and the same sources of error apply.
- Parent (superset) concepts which benefited from specializing on their most frequently occurring child concepts.
- Concepts that singularly, or mutually, benefited from detecting the related concept and thus implicitly a different feature set. These concepts may be entirely disjoint, or part of an ancestor-descendent relationship. This category is the most interesting, as one would not associate some of the disjoint concepts together because of their different semantic meanings.

4.9 Chi-square Conclusion

Chi-square is a powerful method because it discovers concept relationships which are not immediately apparent from their semantic meaning. In contrast, the relationships used for Ancestor boosting are taken straight from an ontology. The two methods are complementary however. Most of the relationships discovered in Ancestor boosting can also be seen in the chi-square results. Some, such as {desert, outdoor} are not.

The chi-square determines concept relationships from the ground-truth annotations. This does make the assumption that the training set is representative of the larger dataset, lest the incorrect relationships be made. In contrast Ancestor boosting is dependent only on an ontology, which is independent of the training set, as it is determined only based on the semantic meanings of the various concepts. Nonetheless, more concepts that result in a MAP gain are identified with chi-square than with Ancestor boosting, a total of 22 for chi-square compared with 17 for Ancestor boosting.

Chi-square is sensitive to variations in the size of the partitions used to divide the ground-truth annotations of each concept. A number of factors influence the choice of partition size, and

more research needs to be done on the choice of partition size and significance level in order to determine the best values that return the largest amount of concept pairs while minimizing the amount of irrelevant pairs.

Another matter for future consideration is replacing the concept detector interpolation procedure with a SVM classifier. As previously stated in 4.3, there is a risk that the EM algorithm for the λ parameter training over-trains, and that a SVM classifier would be more robust under different testing conditions.

Chapter 5

Conclusion

The literature survey that was the formative part of this thesis covered the early beginnings of semantic concept detectors up to the state of the art systems and techniques used today. This served to introduce three techniques, which further improve modern concept detectors, based on the exploitation of the inter-conceptual relationships between the various semantic concepts within a dataset.

The first two techniques, Sibling-confusion removal and Ancestor boosting, employ an ontology to determine the relationships between the concepts of a dataset.

Sibling concepts are semantically very related, and this tends to be reflected in their feature sets, which also are likely to be very similar. This causes confusion among their concept detectors, as they misclassify data into one semantic class while in reality the data belongs to a different, mutually exclusive, sibling class. The Sibling-confusion removal technique reduces the amount of false positives detected by normalizing detector scores based on the confusion factor, a number that indicates the likelihood of a false positive occurring for a particular shot. When run on the MediaMill dataset, Sibling-confusion removal resulted in improved MAP scores for 30 out of 64 concepts. Although the technique showed some promise, it was sensitive to a number of negative influences. This technique was hampered by a poor set of sibling concept groupings provided by the ontology. These groupings have to contain concepts which are semantically related, closed, and where, for every shot, each concept is complementary to the set of remaining concepts, which was not always the case. Next, there is a basic level of performance required by the concept detectors of a sibling set before any positive improvement is noticeable. The effect of weak concept detectors was compounded by the application of Sibling-confusion removal. Finally, there have to be a certain amount of false positives in the concept set under consideration for the technique to be worthwhile. In many cases, this was not so, and the application of the technique led to a decrease in MAP.

Sibling-confusion removal can easily be improved by refining the ontology on which it depends. A human operator can better order the concepts within the ontology to create the correct sibling concept groupings on which this technique depends. Since Sibling-confusion removal runs in linear time with respect to the size of the dataset, and executes quickly, a trial-and-error approach can be used to determine the concepts that benefit from the application of this technique.

The premise of the Ancestor boosting technique is that ancestor concepts occur more often in the dataset, and thus have more robust detectors, than child concepts. Because ancestor concepts are supersets of child concepts, and thus should occur together, it is possible to improve child concept detectors by interpolating them with their ancestor concept detectors, thereby increasing the decision boundaries of the child concept detectors. When applied to the dataset, Ancestor boosting resulted in improved MAP scores for 17 out of 61 distinct concepts. The failure to improve certain concept detectors was due to the semantic distance between the child and ancestor concepts, which meant that there often was little correlation between ancestor and child concepts. Another reason was that the notion that ancestor concept detectors were more robust than child concept detectors proved to be incorrect.

Further improvements for this technique could be achieved performing ancestor boosting with a SVM classifier, instead of linearly interpolating the detectors. That is, the SVM classifier

would take the outputs of the child and ancestor classifiers, and internally generate a combinatory classifier. There is a risk that the EM algorithm for the λ parameter estimation over-trains, and that a SVM classifier would be more robust under different testing conditions.

Chi-square boosting utilizes the chi-square test to identify concepts with similar frequency distributions from a sample of the dataset. This is used to infer concepts that frequently occur together. Detector performance is increased because detectors can make use of the presence of one concept to infer the presence of the related concept. Chi-square boosting resulted in an improvement in 22 out of 36 concepts. Comparatively, this is a greater increase than delivered by the Ancestor boosting and Sibling-confusion removal techniques. The concept improvements could be attributed to: child concepts leveraging the functionality of their more powerful ancestor concept detectors (identical to Ancestor boosting), parent concepts which specialized on their most frequently occurring child concepts, and concepts which singly or mutually benefited from detecting the related concept and thus implicitly a different feature set.

Chi-square does not depend on an ontology to determine concept relationships. It does, however, assume that the training set is a representative sample of the overall dataset when determining the various concept relationships. Chi-square is also sensitive to variations in the significance level and partition size parameters when performing the chi-test to determine concept relationships. This should be further explored in order to determine whether additional concept relationships could be discovered that would lead to further concept boosting. The interpolation procedure that creates new concept detectors is identical to the one in Ancestor boosting, and for like reasons it should be determined whether an SVM classifier should replace this interpolation procedure.

A further direction of research is to investigate the effects of combining the three techniques presented. Ancestor boosting and Chi-square boosting are very similar in nature, so the logical approach would be to take the union of the concept detectors resulting from these two techniques, prior to applying Sibling-confusion removal. It is not possible to applying Sibling-confusion removal first, as the re-ranking procedure breaks the confidence metric. A brief inquiry suggests that the above combination results in additional MAP improvements, better than the best single boosting technique. A table with some preliminary findings follows:

Table: Combined boosting

Concept	Original MAP	Best Ancestor or Chi boosting MAP	Sibling-confusion removal MAP	Best single detector % change	Ancestor or Chi Boosting followed by Sibling-confusion removal MAP	Combined % change
Charts	0.2541	0.2658	0.2697	6.13%	0.3049	19.98%
Maps	0.3039	0.3383	0.2989	11.32%	0.2878	-5.31%
Tower	0.0235	0.0254	0.0223	8.07%	0.0255	8.46%
House	0.0066	0.0080	0.0066	21.15%	0.0081	21.37%
Government building	0.0793	0.0793	0.0793	0.02%	0.0794	0.09%

There are too few concepts in the current dataset however, which have both MAP improvements, from Ancestor or Chi-square boosting, and which also are sibling concepts, to make informed judgments about the effectiveness of combining these techniques.

References

[1] W. H. Adams, G. Iyengar, C-Y. Lin, M. R. Naphade, C. Neti, H.J. Nock, J. R. Smith, “Semantic Indexing of Multimedia Content using Visual, Audio and Text cues”, EURASIP Journal on Applied Signal Processing, 2003, pp. 170-185.

[2] A. Alatan, A. Akansu, and W. Wolf, “Multi-modal Dialog Scene Detection Using Hidden Markov Models for Content-Based Multimedia Indexing”, Multimedia Tools and Applications, Kluwer Academic Publishers, 2001, pp. 137-151.

[3] A. Amir, M. Berg, S-F. Chang, W. Hsu, G. Iyengar, C-Y. Lin, M. Naphade, A. Natsev, C. Neti, H. Nock, J. R. Smith, B. Tseng. Y. Wu, and D. Zhang, “IBM Research TRECVID-2003 Video Retrieval System”, TRECVID, 2003,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[4] A. Amir, J. Argillander, M. Berg, S-F. Chang, M. Franz, W. Hsu, G. Iyengar, J. R. Kender, L. Kennedy, C-Y. Lin, M. Naphade, A. Natsev, J. R. Smith, J. Tesic, G. Wu, R. Yan, and D. Zhang, “IBM Research TRECVID-2004 Video Retrieval System”, TRECVID, 2004,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[5] A. Amir, J. Argillander, M. Campbell, A. Haubold, G. Iyengar, S. Ebadollahi, F. Kang, M. Naphade, A. Natsev, J. R. Smith, J. Tesic, and T. Volkmer, “IBM Research TRECVID-2005 Video Retrieval System”, TRECVID, 2005,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[6] S. Ayache, G. Quenot, J. Gensel, and S. Satoh, “Using Topic Concepts for Semantic Video Shots Classification”, International Conference on Image and Video Retrieval CIVR'06, Tempe, USA, July, 2006.

[7] N. Babaguchi and N. Nitta, “Intermodal Collaboration: A Strategy for Semantic Content Analysis for Broadcasted Sports Video”, ICIP, 2003.

[8] R. Benmokhtar, E. Dumont, B. Huet and B. Merialdo, “Eurecom at TRECVID 2006: Extraction of High-Level Features and BBC Rushes Exploitation”, TRECVID, 2006,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[9] M. Bertini, R. Cucchiara, A. Del Bimbo, and C. Torniai, “Automatic Video Annotation using Ontologies Extended with Visual Information”, ACM Multimedia, Singapore, 2005.

[10] H. Blanken(ed.) et al. “Multimedia Retrieval”, p. 63-66, March 13, 2007.

[11] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, Data Mining and Knowledge Discovery, 2, p. 121-167, Kluwer Academic Publishers, Boston, 1998.

[12] C-C Chang and C-J Lin, “LIBSVM : a library for support vector machines, 2001”,

Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[13] S-F. Chang, W. Hsu, L. Kennedy, L. Xie, A. Yanagawa, E. Zavesky, and D-Q. Zhang, “Columbia University TRECVID-2005 Video Search and High-Level Feature Extraction”, TRECVID, 2005,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[14] S-F. Chang, W. Hsu, W. Jiang, L. Kennedy, D. Xu, A. Yanagawa, and E. Zavesky, “Columbia University TRECVID-2006 Video Search and High-Level Feature Extraction”, TRECVID, 2006,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[15] J. Fan, W. G. Aref, A. K. Elmagarmid, M-S. Hacid, M. S. Marzouk, X. Zhu, "Multiview: Multilevel video content representation and retrieval", Journal of Electronic Imaging, October 2001, pp. 895-908.

[16] J. Fan, H. Luo, and X. Lin, "Semantic Video Classification by Integrating Flexible Mixture Model with Adaptive EM Algorithm", SIGMM international workshop on Multimedia information retrieval, ACM Press, Berkely, USA, 2003, pp. 9-16

[17] J. Fan, H. Luo, X. Lin, and L. Wu, "Semantic Video Classification and Feature Subset Selection under Context and Concept Uncertainity", Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, Tuscon, USA, 2004, pp. 192-201.

[18] J. C. van Gemert, J-M. Geusebroek, C. J. Veenman, C. Snoek, and A. Smeulders, "Robust Scene Categorization by Learning Image Statistics in Context", Computer Vision and Pattern Recognition Workshop, 2006.

[19] H. Gunes and M. Piccardi, "Affect Recognition from Face and Body: Early Fusion vs. Late Fusion", IEEE System Man and Cybernetics, 2005.

[20] N. Haering, R. Qian, M. Sezan, "A Semantic Event Detection Approach and Its Application to Detecting Hunts in Wildlife Video", IEEE Transaction on Circuits and Systems for Video Technology, September 2000.

[21] A. Hauptmann, R. V. Baron, M-Y. Chen, M. Christel, P. Duygulu, C. Huang, R. Jin, W-H. Lin, T. Ng, N. Moraveji, N. Papernick, C. Snoek, G. Tzanetakis, J. Yang, R. Yan, and H. D. Wactlar, "Informedia at TRECVID 2003: Analyzing and Searching Broadcast News Video", TRECVID, 2003,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[22] A. Hauptmann, R. V. Baron, M. Christel, R. Conescu, J. Gao, Q. Jin, W-H. Lin, J-Y. Pan, S. M. Stevens, R. Yan, J. Yang, and Y. Zhang, "CMU's Informedia's TRECVID 2005 Skirmishes", TRECVID, 2005,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[23] A. Hauptmann, M-Y. Chen, M. Christel, W-H. Lin, R. Yan, and J. Yang, "Multi-Lingual Broadcast News Retrieval", TRECVID, 2006,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[24] M. Hearst (ed) "Support Vector Machines", IEEE Intelligent Systems, July/August 1998. (svmOverview.pdf)

[25] C-W Hsu, C-C Chang, and C-J Lin, "A Practical Guide to Support Vector Classification",

http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[26] G. Iyengar, H.J. Nock, and C. Neti, "Discriminative Model Fusion for Semantic Concept Detection and Annotation in Video", ACM Multimedia, Berkely, USA, 2003.

[27] W. Jiang, S-F Chang, and A. C. Loui, "Active Context-Based Concept Fusion with Partial User Labels", ICIP, 2006

[28] T. Joachims, "Making large-scale SVM Learning Practical. Advances in Kernel Methods-Support Vector Learning", B. Scholkopf, C. Burges, and A. Smola (ed.), MIT-Press, 1999.

[29] W-H. Lin, R. Jin, A. Hauptmann, "Meta-classification of Multimedia Classifiers", First International Workshop in Multimedia and Complex Data, Taipei, Taiwan, 2002.

[30] W-H. Lin and A. Hauptmann, "News Video Classification Using SVM-based Multimodal Classifiers and Combination Strategies", ACM Multimedia, Juan-les-Pins, France, 2002.

[31] The Lowlands Team, "Lazy Users and Automatic Video Retrieval Tools in the Lowlands", Tenth Text Retrievval Conference (TREC-10), NIST Special Publication, 2001.

[32] LSCOM Lexicon Definitions and Annotations Version 1.0, DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia, Columbia University ADVENT Technical Report #217-2006-3, March 2006.

[33] H. Luo and J. Fan, "Building Concept Ontology for Medical Video Annotation", ACM Multimedia, Santa Barbara, 2006.

[34] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Y. Ng, "Improving Text Classification by Shrinkage in a Hierarchy of Classes", Proceedings of ICML-98, 15th International Conference on Machine Learning, 1998.

[35] M. Naphade, L. Kennedy, J. R. Kender, S-F Chang, J. R. Smith, P. Over, A. Hauptmann, "A light Scale Concept Ontology for Multimedia Understanding for TRECVID 2005", IBM Technical Report, 2005.

http://www.ee.columbia.edu/ln/dvmm/lscom/ibmtr2005-lscom.pdf

[36] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years", IEEE Transactions on Pattern Analysis and Machine Intelligence, December 2000.

[37] M. Sewell, "SVM-Dark"

http://www.cs.ucl.ac.uk/staff/M.Sewell/svmdark/

[38] J. Smith, A. Jaimes, C-Y Lin, M. Naphade, A. Natsev, B. Tseng, "Interactive Search Fusion Methods for Video Database Retrieval", ICIP, 2003.

[39] J. Smith, M. Campbell, M. Naphade, A. Natsev and J. Tesic, "Learning and Classification of Semantic Concepts in Broadcast Video", Conf. on Intelligence Analysis, 2005.

[40] C. Snoek and A. Hauptmann, "Learning to Identify TV News Monologues by Style and Context", Carnegie Mellon University, 2003.

[41] C. Snoek, M. Worring, and A. Smeulders, "Early Versus Late Fusion in Semantic Video Analysis", ACM Multimedia, Singapore, 2005.

[42] C. Snoek, M. Worring, "A State-of-the-art Review on Multimodal Video Indexing", Proceedings of the 8th Annual Conference of the Advanced School for Computing and Imaging, Lochem, the Netherlands, 2002.

[43] C. Snoek, M. Worring, J. C. van Gemert, J-M. Geusebroek, and A. Smeulders, "The Challenge Problem for Automatic Detection of 101 Semantic Concepts in Multimedia", ACM Multimedia, Santa Barbara, USA, 2006.

[44] C. Snoek and M. Worring, "Multimodal Video Indexing: A Review of the State-of-the-art", Intelligent Sensory Information Systems Group, University of Amsterdam, The Netherlands, 2001.

[45] C. Snoek and M. Worring, "Multimodal Video Indexing: A Review of the State-of-the-art", Multimedia Tools and Applications, Springer Science, The Netherlands, 2005.

[46] C. Snoek and M. Worring, "A Review of Multimodal Video Indexing", Intelligent Sensory Information Systems Group, University of Amsterdam, The Netherlands, 2002.

[47] C. Snoek and M. Worring, "Multimedia Event-Based Video Indexing Using Time Intervals", IEEE Transactions on Multimedia, August 2005.

[48] C. Snoek, M. Worring, and A. Hauptmann, "Learning Rich Semantics from News Video Archives by Style Analysis", TOMCCAP, ACM Press, New York, USA, 2006.

[49] C. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring, "Adding Semantics to Detectors for Video Retrieval", IEEE Transactions on Multimedia, 2006

[50] C. Snoek, M. Worring, J-M. Geusebroek, D. C. Koelma, and F. J. Seinstra, "The MediaMill TRECVID 2004 Semantic Video Search Engine", TRECVID, 2004,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[51] C. Snoek, J. C. van Gemert, J-M. Geusebroek, B. Huurnink, D. C. Koelma, G. P. Nguyen, O. de Rooij, F. J. Seinstra, A. Smeulders, C. J. Veenman, and M. Worring, "The MediaMill TRECVID 2005 Semantic Video Search Engine", TRECVID, 2005,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[52] C. Snoek, J. C. van Gemert, T. Gevers, B. Huurnink, D. C. Koelma, M. van Liempt, O. de Rooij, K. van de Sande, F. J. Seinstra, A. Smeulders, A. Thean, C. J. Veenman, and M. Worring, "The MediaMill TRECVID 2006 Semantic Video Search Engine", TRECVID, 2006,

[53] D. Song, H. T. Liu, M. Cho, H. Kim, and Pankoo Kim, "Domain Knowledge Ontology Building for Semantic Video Event Description", CIVR, 2005.

[54] S. Tsekeridou and I. Pitas, "Content-Based Video Parsing and Indexing Based on Audio-Visual Interaction", IEEE Transaction on Circuits and Systems for Video Technology, April 2001.

[55] B. L. Tseng, C-Y. Lin, M. Naphade, A. Natsev, and J. Smith, "Normalized Classifier Fusion for Semantic Visual Concept Detection", ICIP, 2003.

[56] VideoAnnex, http://www.research.ibm.com/VideoAnnEx/

[57] T. Westerveld, T. Ianeva, L. Boldareva, A. P. de Vries, and D. Hiemstra, "Combining Information Sources for Video Retreival: The Lowlands Team at TRECVID 2003", TRECVID, 2003,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[58] T. Westerveld, J. C. van Gemert, R. Cornacchia, D. Hiemstra, and A. P. de Vries, "An Integrated Approach to Text and Image Retrieval", TRECVID, 2005,

http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html

[59] T. Westerveld, A. P. de Vries, A. van Ballegooij, F. de Jong, and D. Hiemstra, "A Probabilistic Multimedia Retrieval Model and its Evaluation", EURASIP Journal on Applied Signal Processing, special issue on Unstructured Information Management from Multimedia Data Sources, 2003, pp. 186-198.

[60] T. Westerveld and A. P. de Vries, "Generative probabilistic models for multimedia retrieval: query generation versus document generation", IEEE Vision Image and Signal Processing, December 2005, pp. 852-858.

[61] T. Westerveld, "Using generative probabilistic models for multimedia retrieval", Phd. Thesis, Enschede, The Netherlands, 2004.

[62] Y. Wu, B. Tseng, and J. R. Smith, "Ontology-based Multi Classification Learning For Video Concept Detection", IEEE International Conference on Multimedia and Expo, ICME, 2004.

[63] Y. Wu, E. Chang, K.-C. Chang, and J. Smith, "Optimal Multimodal Fusion for Multimedia Data Analysis", ACM Multimedia, New York, USA, 2004.

[64] H. Xu and T-S. Chua, "Fusion of AV Features and External Information Sources for Event Detection in Team Sports Video", TOMCCAP, 2006.

[65] R. Yan and A. Hauptmann, "The Combination Limit in Multimedia Retrieval", ACM Multimedia, Berkely, USA, 2003.

[66] R. Yan, J. Yang, and A. Hauptmann, "Learning Query-Class Dependent Weights in Automatic Video Retrieval", ACM Multimedia, New York, USA, 2004.

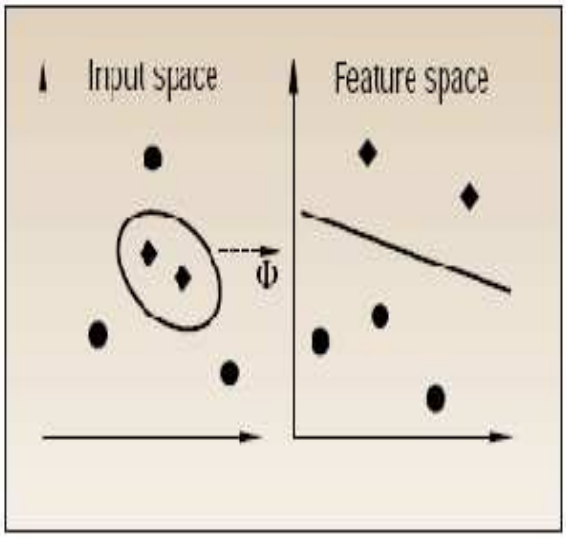
[67] R. Yan, M. Chen, and A. Hauptmann, "Mining Relationship Between Video Concepts Using Probabilistic Graphical Models", IEEE International Conference on Multimedia and Expo, Toronto, Canada, 2006.

[68] R. Zhang, R. Sarukkai, J-H. Chow, Wei Dai, Z. Zhang, “Joint Categorization of Queries and Clips for Web-based Video Search”, ACM Multimedia, Santa Barbara, USA, 2006.

Annex 1

SVM Theory

Given training vectors (i.e. low-level graphical or textual features) x_i and their associated class labels (i.e. whether a semantic concept in the shot, true or false) y_i, (x_i, y_i) for i=1, 2…L where L is the number of training samples, and labels $y_i \in \{1,-1\}$, a SVM must construct a model from this training set which will allow it to predict unlabeled vectors in the future. It does so by applying a function ϕ, which maps the training vectors x_i into a higher dimensional space, whereupon the SVM finds a linearly separating hyperplane that maximally separates the two classes. [24, 25] The intuitive idea behind the mapping is to ensure that there is an easy, linearly separable classification of the training set in the higher dimension, even if this was not the case in the dimension of the input space.



Source: [24]

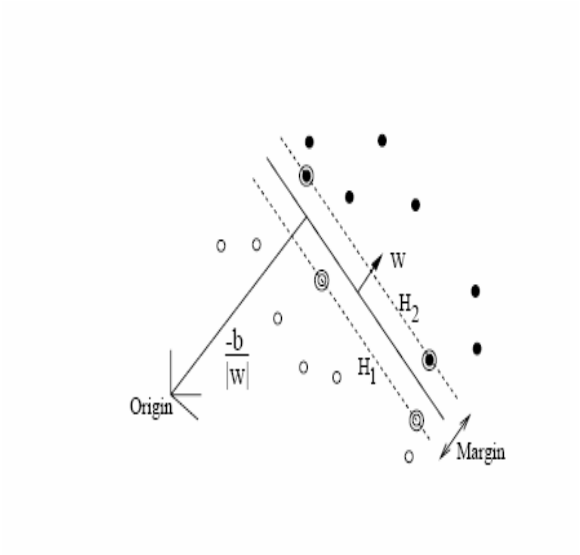
This transformation also reduces the risk of curse of dimensionality [63], and the maximally separating hyper plane prevents overtraining.

The formula of a hyper plane is given by x.w+b=0 where w is the normal vector, |b|/||w|| the perpendicular distance from hyperplane to the origin, and ||w|| the Euclidian norm of w. Given a hyperplane, x.w +b, which separates the two classes, define d+ and d- as the shortest distance from the hyperplane to the closest positively and closest negatively labeled examples. Then the margin of the hyperplane can be given as $d_+ + d_-$. [11] The SVM algorithm seeks to find the hyperplane with the largest margin, intuitively because this ensures the data is as 'far' away as possible from the decision boundary, thus minimizing the risk of over fitting the data and of misclassifying future examples.

Thus for the linearly separable case, the constraints on the training data can be formulated as follows:

I $x_i \cdot w+b \geq +1$ for $y_i=+1$
II $x_i \cdot w+b \leq -1$ for $y_i=-1$, which combined is written as:
III $y_i(x_i \cdot w+b)-1 \geq 0$ for all i

The points for which equality I holds lie on hyperplane $H_1 : x_i \cdot w+b=1$, with normal w and perpendicular distance from the origin |1-b|/||w||.
The points for which equality II holds lie on hyperplane $H_2 : x_i \cdot w+b=-1$, with normal w and perpendicular distance from the origin |-1-b|/||w||.
Hence the margin= $d_+ - d_- = (1-b)/||w|| - (-1-b)/||w|| = 2/||w||$

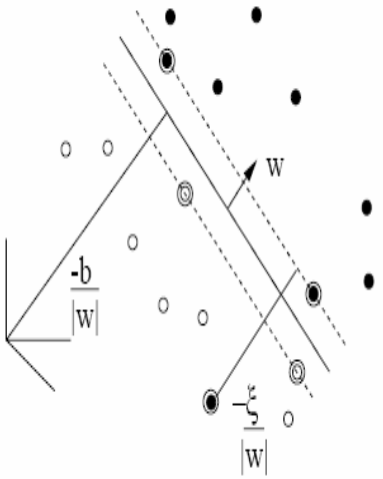


Source: [11]

In order to maximize the margin, $||w||/2 = w^T w/2$ must be minimized, subject to the constraints $y_i(x_i \cdot w+b)-1 \geq 0$ for all i [11]. Therefore the problem of finding the optimal hyperplane is a constrained quadratic optimization problem, which is solvable in polynomial time [24]. Training points which satisfy equality III, and thus lie on either of the hyper planes H_1 or H_2, are called support vectors because they actively affect the solution. These points are marked with an extra circle. The above proof does not consider the non-separable case, which could occur with erroneous training points or true positives, outliers on the far side of the decision boundary.

Nonetheless, a linearly separating hyperplane can be constructed by introducing a slack variable ξ_i. The minimization problem then becomes: $w^T w/2 + C_i=1\Sigma L\ \xi_i$ subject to the constraints $y_i(x_i.w+b)-1 +\xi_i \geq 0\ \forall_i$, $\xi_i \geq 0$ [25] . Thus positive values $\xi_i > 0$ correspond to training examples that have violated the constraints, either they are misclassified or they are correctly classified and fall within the margin.

C is a parameter, chosen by the user, which assigns a penalty to error. Outliers can be considered support vectors that contribute to the placement of the hyper plane-decision boundary. As such, C can be varied to affect their influence. Erroneous seeming data points can be penalized and have their influence reduced on the selection of the hyper plane-decision boundary. This would maximize the margin, and would correspond to selecting a low C value, thus moving the hyperplane away from the outliers. If instead, outliers were to be favored, as would be the case if they were treated as true positive classifications, a high C value ought to be chosen. This would move the hyperplane closer to the outliers, thereby minimizing training error but also resulting in a decreased margin. [10]



Source: [11]

Earlier it was stated that SVMs map input data into a much higher dimensional feature space, where it would be much more likely that the data was linearly separable.

This mapping is done by a function, ϕ and the SVM conditions become:

Minimize w,b, ξ : $w^T w/2 + C_i=1\Sigma L\ \xi_i$;

subject to the constraints y_i ($\phi(x_i).w+b$)-1 $+\xi_i \geq 0 \; \forall_i$, $\xi_i \geq 0$ [25]

An important point is that computationally, the data does not explicitly have to be mapped to the higher-dimensional space which is an expensive operation, rather computing the dot product in this space is sufficient.[10, 24, 25] As such, SVMs use kernel functions of the form: $K(x_i,x_j)= \phi(x_i^T)\phi(x_j)$ to perform this transformation.

Kernels currently in use are:

linear: $K(x_i, x_j) = x^T_i x_j$

polynomial: $K(x_i, x_j) = (\gamma x^T_i x_j + r)^d$, $\gamma > 0$.

radial basis function (RBF): $K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$, $\gamma > 0$. ($\gamma = 1/(2\sigma^2)$),

sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$. [24, 25]

where *r, d,* and γ are kernel parameters that are experimentally determined through cross-validation during the SVM learning stage.

The sigmoid kernel trains a SVM as a 3-layer neural network. [24][11] For general purpose classification, both the RBF [25] and polynomial [10] kernels are recommended. Hsu et al. further endorse the RBF kernel because it is a non-linear mapping, thus allowing for training data which is not separable in the input domain. Also, the RBF kernel supersedes the linear kernel, as the linear kernel is a special case of a RBF kernel. They also argue that the RBF kernel is simpler to train than the polynomial kernel, as it has less kernel parameters, one as opposed to two. Finally they argue that RBF kernels do not suffer from numerical difficulties, $0<K_{ij}<1$, whereas the polynomial kernel $(\gamma x^T_i x_j + r>1)^d$ may go to infinity, or zero $(\gamma x^T_i x_j + r<1)^d$, for large d. Likewise they argue that the sigmoid kernel may not be valid for certain parameters. [25]

During SVM training with the RBF kernel, the parameters γ, C, and the positive and negative weights of training examples may be varied to produce a different model. The γ and C have been discussed thoroughly above, but the training weights have not been mentioned. These weights are used in cases where there are few positive training examples. By assigning a higher weight to positive examples, the SVM is forced to include the positive instances of a class in determining the demarcating hyperplane. This is necessary for cases with a significantly larger number of negative examples; where if the positive weight is not set, the SVM may optimize on the many negative examples and create a decision boundary unable to detect positive instances. It may be trivially accurate as it is able to classify negative instances with a high degree of certainty, but it isn't able to detect the few, relevant, positive cases of the class at all.

Annex 2

Sibling-confusion removal results

The following table shows the sibling concepts present in the ontology, their original MAP, and the updated MAP after the confusion removal algorithm had been run. Their parent concept is listed as well, or Root, if they had none. Some sibling sets had additional concepts added, labeled superset, or concepts removed, labeled subset. This was done because the operation seemed logical from a semantic perspective, in order to gain insight into whether it affected the MAP.

Table 2 Sibling-confusion removal results				
Parent class	**Concept**	**Original MAP**	**Updated MAP**	**Performance change**
Studio	splitscreen	0.3210	0.3373	5.08%
Root (subset) 'location'	snow	0.0452	0.0472	4.42%
Studio	newspaper	0.1212	0.1263	4.20%
Face	bush_jr	0.0396	0.0407	2.70%
Sports	soccer	0.0793	0.0814	2.59%
Face	table	0.0375	0.0385	2.57%
Face	kerry	0.0022	0.0022	2.00%
Original indoor	outdoor	0.7095	0.7212	1.65%
Studio	weather	0.7068	0.7181	1.60%
Root node	indoor	0.5926	0.6019	1.56%
Face	clinton	0.1894	0.1923	1.54%
Face	powell	0.0849	0.0859	1.21%
Face	tony_blair	0.0147	0.0148	0.89%
Vehicle	car	0.2458	0.2479	0.85%
Root(subset) 'program type'	Drawing_cartoon	0.1811	0.1823	0.66%
Root(subset) 'program type'	Drawing_cartoon	0.1811	0.1823	0.62%
Animal	dog	0.1027	0.1034	0.61%
government leader (superset)	hassan_nasrallah	0.0044	0.0045	0.46%
government leader (superset)	bush_sr	0.0001	0.0001	0.31%
Face	arrafat	0.0342	0.0343	0.25%
Face	bush_sr	0.0001	0.0001	0.13%
Face	sharon	0.0348	0.0348	0.13%
Building	house	0.0066	0.0066	0.11%
Face	monologue	0.0736	0.0736	0.08%
Face	hassan_nasrallah	0.0044	0.0045	0.07%
Water body (superset)	swimming pool	0.0014	0.0014	0.03%
Root(subset) 'program type'	Entertainment	0.2565	0.2566	0.03%
Building	govt. building	0.0793	0.0793	0.02%
Water body (superset)	waterfall	0.4152	0.4153	0.02%
Water body	waterfall	0.4152	0.4153	0.02%
Face	duo_anchor	0.1080	0.1080	0.01%
Animal	fish	0.4075	0.4076	0.01%
Original drawing	drawing	0.0440	0.0440	0.01%

Water body (superset)	river	0.6540	0.6540	0.00%
Sports	cycling	0.8875	0.8875	0.00%
Face	allawi	0.0022	0.0022	0.00%
government leader (superset)	kerry	0.0022	0.0022	0.00%
explosion	nightfire	0.2489	0.2489	0.00%
explosion	candle	0.0801	0.0801	-0.01%
Water body	swimming pool	0.0014	0.0014	-0.01%
Root (subset) ‘location’	studio	0.6653	0.6652	-0.02%
Water body (superset)	beach	0.0652	0.0652	-0.03%
Water body	beach	0.0652	0.0652	-0.03%
Root(subset) ‘program type’	Entertainment	0.2565	0.2564	-0.04%
government leader (superset)	arrafat	0.0342	0.0341	-0.14%
Face	anchor	0.6192	0.6172	-0.33%
People	male	0.0678	0.0676	-0.38%
Sports	football	0.0197	0.0196	-0.38%
Face	lahoud	0.1151	0.1146	-0.47%
government leader (superset)	lahoud	0.1151	0.1145	-0.55%
Root (subset) ‘location’	Mountain	0.0918	0.0912	-0.60%
Root (subset) ‘location’	Desert	0.0933	0.0927	-0.66%
government leader (superset)	powell	0.0849	0.0842	-0.83%
Studio	anchor	0.6192	0.6138	-0.88%
Face	hu_jintao	0.0436	0.0432	-0.95%
government leader (superset)	bush_jr	0.0396	0.0392	-1.01%
Sports	golf	0.0424	0.0419	-1.09%
Root(subset) ‘program type’	weather	0.7068	0.6974	-1.34%
Vehicle	truck	0.0418	0.0411	-1.53%
Animal	horse	0.0003	0.0003	-1.60%
Face	splitscreen	0.3210	0.3158	-1.61%
government leader (superset)	allawi	0.0022	0.0022	-1.67%
government leader (superset)	hu_jintao	0.0436	0.0428	-1.93%
Animal	bird	0.7611	0.7442	-2.23%
Vehicle	bus	0.0088	0.0086	-2.25%
government leader (superset)	clinton	0.1894	0.1848	-2.41%
government leader (superset)	tony_blair	0.0147	0.0143	-2.52%
Sports	tennis	0.2985	0.2905	-2.68%
Vehicle	aircraft	0.1147	0.1115	-2.82%
Original drawing	cartoon	0.2783	0.2683	-3.57%
Face	male	0.0678	0.0654	-3.57%
Sports	basketball	0.1791	0.1720	-3.94%
Face	female	0.0610	0.0579	-5.11%
Studio	duo_anchor	0.1080	0.1024	-5.17%
Building	tower	0.0235	0.0223	-5.26%
Root (subset) ‘location’	water body	0.1317	0.1213	-7.93%
Face	government_leader	0.2218	0.2000	-9.83%
Vehicle	tank	0.0107	0.0097	-10.06%
Vehicle	boat	0.0834	0.0714	-14.34%
government leader (superset)	sharon	0.0348	0.0280	-19.41%
Root(subset) ‘program type’	Sports	0.2307	0.1678	-27.28%
Root(subset) ‘program type’	Sports	0.2307	0.1667	-27.78%
Sports	racing	0.1754	0.1261	-28.08%

Root (subset) ‘location’	urban	0.1948	0.1351	-30.63%
People	female	0.0610	0.0406	-33.44%
Vehicle	bicycle	0.2234	0.1223	-45.25%

Annex 3

Ancestor boosting results

The original MAP score for the child concept is listed, followed by the MAP score after ancestor boosting. Some concepts in the training set did not have scores in the full range [0-1], and were first normalized for compliance. This affected the λ values, and changed the resultant MAP scores, which is the last category on display. For comparison purposes, sometimes an ancestor concept was omitted from a sub tree. The child concept is the leftmost element in each set, and ancestors are listed in ascending order.

Table 3 Ancestor boosting results

Sub-tree	Concept	Original MAP	Boosted MAP	Boosted and Normalized MAP	Performance change
swimming pool, water body	swimming pool	0.0014	0.0054	0.0054	285.54%
tank, vehicle	tank	0.0107	0.0300	0.0250	132.91%
desert, outdoor	desert	0.0933	0.1837	0.1837	96.87%
swimming pool, water body, outdoor	swimming pool	0.0014	0.0021	0.0021	48.04%
basketball, sports	basketball	0.1791	0.2207	0.2207	23.26%
bus, vehicle	bus	0.0088	0.0117	0.0108	21.89%
house, building	house	0.0066	0.0080	0.0080	21.15%
maps, graphics	maps	0.3039	0.3383	0.3383	11.32%
explosion, violence	explosion	0.0782	0.0857	0.0857	9.61%
tower, building	tower	0.0235	0.0239	0.0254	8.07%
tennis, sports	tennis	0.2985	0.3184	0.3184	6.65%
charts, graphics	charts	0.2541	0.2657	0.2658	4.58%
anchor, studio	anchor	0.6192	0.6358	0.6358	2.69%
anchor, studio, indoor	anchor	0.6192	0.6355	0.6355	2.62%
anchor, indoor	anchor	0.6192	0.6334	0.6334	2.29%
horse_racing, horse	horse_racing	0.0003	0.0003	0.0003	0.71%
flag_usa, flag	flag_usa	0.1568	0.1573	0.1573	0.31%
night fire, explosion	night fire	0.2489	0.2495	0.2495	0.25%
studio, indoor	studio	0.6653	0.6657	0.6657	0.05%
face, people	face	0.8921	0.8922	0.8922	0.01%
cycling, bicycle	cycling	0.8875	0.8875	0.8875	0.00%
allawi, government leader, face	allawi	0.0002	0.0002	0.0002	0.00%
drawing, drawing_cartoon	drawing	0.0440	0.0440	0.0440	0.00%
cartoon, drawing_cartoon	cartoon	0.2783	0.2783	0.2783	0.00%
fish, animal	fish	0.4075	0.4066	0.4066	-0.22%
car, vehicle	car	0.2458	0.2432	0.2432	-1.08%
football, sports	football	0.0197	0.0194	0.0194	-1.19%
basketball, walking_running	basketball	0.1791	0.1753	0.1765	-1.47%
bird, animal	bird	0.7611	0.7466	0.7466	-1.92%
candle, explosion	candle	0.0801	0.0784	0.0784	-2.16%

urban, outdoor	urban	0.1948	0.1877	0.1877	-3.65%
golf, sports	golf	0.0424	0.0404	0.0404	-4.70%
bush_jr, government leader, face	bush_jr	0.0396	0.0375	0.0375	-5.30%
soccer, sports	soccer	0.0793	0.0745	0.0745	-6.07%
anchor, face	anchor	0.6192	0.5714	0.5714	-7.72%
military, walking_running	military	0.2370	0.2370	0.2182	-7.93%
split screen, indoor	split screen	0.3210	0.2950	0.2954	-7.96%
house, building, outdoor	house	0.0066	0.0061	0.0061	-8.82%
splitscreen, studio, indoor	split screen	0.3210	0.2923	0.2926	-8.85%
smoke, violence	smoke	0.3659	0.3323	0.3324	-9.15%
cloud, sky	cloud	0.0785	0.0710	0.0710	-9.53%
river, waterbody	river	0.6540	0.5872	0.5872	-10.20%
boat, vehicle	boat	0.0834	0.0747	0.0747	-10.41%
weather, indoor	weather	0.7068	0.6024	0.6024	-14.78%
weather, studio, indoor	weather	0.7068	0.6013	0.6013	-14.93%
horse_racing, horse, animal	horse_racing	0.0003	0.0002	0.0002	-15.58%
truck, vehicle	truck	0.0418	0.0348	0.0348	-16.65%
cloud, sky, outdoor	cloud	0.0785	0.0612	0.0612	-22.05%
aircraft, vehicle	aircraft	0.1147	0.0880	0.0879	-23.40%
Original arrafat	arrafat	0.0342	0.0259	0.0261	-23.53%
female, face	female	0.0610	0.0466	0.0466	-23.66%
road, outdoor	road	0.2123	0.1598	0.1599	-24.65%
football, walking_running	football	0.0197	0.0137	0.0137	-30.59%
female, face, people	female	0.0610	0.0408	0.0408	-33.05%
mountain, outdoor	mountain	0.0918	0.0610	0.0610	-33.55%
government leader, face	government leader	0.2218	0.1404	0.1404	-36.71%
tower, building, outdoor	tower	0.0235	0.0145	0.0145	-38.15%
kerry, government leader	kerry	0.0022	0.0013	0.0013	-38.36%
male, face, people	male	0.0678	0.0406	0.0406	-40.07%
tree, outdoor	tree	0.0626	0.0344	0.0344	-45.05%
male, face	male	0.0678	0.0360	0.0360	-46.85%
soccer, walking_running	soccer	0.0793	0.0418	0.0410	-48.30%
male, people	male	0.0678	0.0347	0.0347	-48.82%
female, people	female	0.0610	0.0304	0.0304	-50.13%
cycling, sports	cycling	0.8875	0.3986	0.3986	-55.09%
tennis, walking_running	tennis	0.2985	0.1311	0.1298	-56.52%
car, vehicle, outdoor	car	0.2458	0.0989	0.0989	-59.76%
monologue, overlayed_text	monologue	0.0736	0.0291	0.0291	-60.49%
river, waterbody, outdoor	river	0.6540	0.2532	0.2532	-61.28%
office, indoor	office	0.0452	0.0166	0.0166	-63.25%
monologue, people	monologue	0.0736	0.0251	0.0251	-65.90%
racing, sports	racing	0.1754	0.0577	0.0577	-67.10%
boat, vehicle, outdoor	boat	0.0834	0.0268	0.0268	-67.88%
meeting, indoor	meeting	0.2109	0.0645	0.0645	-69.40%
Original government building	government building	0.0793	0.0246	0.0225	-71.70%
military, walking_running, people	military	0.2370	0.0596	0.0596	-74.85%
table, indoor	table	0.0375	0.0093	0.0093	-75.22%

dog, animal	dog	0.1027	0.0214	0.0214	-79.22%
football, walking_running, people	football	0.0197	0.0040	0.0040	-79.45%
duo_anchor, studio, indoor	duo_anchor	0.1080	0.0213	0.0214	-80.22%
duo_anchor, indoor	duo_anchor	0.1080	0.0207	0.0207	-80.85%
court, indoor	court	0.0297	0.0055	0.0055	-81.45%
night fire, violence	night fire	0.2489	0.0451	0.0452	-81.83%
police_security, walking_running	police_security	0.0825	0.0122	0.0130	-84.21%
prisoner, people	prisoner	0.0508	0.0069	0.0069	-86.35%
beach, waterbody	beach	0.0652	0.0037	0.0082	-87.39%
police_security, walking_running, people	police_security	0.0825	0.0090	0.0090	-89.10%
newspaper, studio, indoor	newspaper	0.2109	0.0127	0.0127	-93.96%
newspaper, indoor	newspaper	0.2109	0.0121	0.0121	-94.27%
soccer, walking_running, people	soccer	0.0793	0.0027	0.0027	-96.55%
basket ball, walking_running, people	basketball	0.1791	0.0054	0.0054	-96.97%
tennis, walking_running, people	tennis	0.2985	0.0043	0.0043	-98.55%
beach, waterbody, outdoor	beach	0.0652	0.0008	0.0008	-98.76%
bicycle, vehicle	bicycle	0.2234	0.2234	0.0010	-99.54%

Annex 4

Chi-square boosting results

The concepts listed were identified using a chi-square test with significance value of 0.05, and evaluated for the ground-truth shots, which were partitioned into blocks of 40. The original MAP score for the each concept is listed, followed by the MAP score after interpolation with the related concept. Some concepts were related to more than one other concept, and detectors were created for the total related set. Some concepts in the training set did not have scores in the full range [0-1], and were first normalized for compliance. This affected the λ values, and changed the resultant MAP scores, which is the last category on display.

Table 3 Chi-square boosting results					
Concept pair	**Concept**	**Original MAP**	**Updated MAP**	**Updated and normalized MAP**	**Performance change**
swimmingpool, duo_anchor	swimming pool	0.0014	0.0034	0.0034	140.07%
bicycle, cycling	bicycle	0.2234	0.3444	0.3444	54.14%
house, building	house	0.0066	0.0080	0.0080	21.15%
maps, graphics	maps	0.3039	0.3383	0.3383	11.32%
tower, building	tower	0.0235	0.0239	0.0254	8.07%
waterbody, beach, waterfall, boat	waterbody	0.1317	0.1382	0.1421	7.86%
waterbody, boat	waterbody	0.1317	0.1382	0.1382	4.94%
people, face	people	0.8897	0.9311	0.9311	4.65%
charts, graphics	charts	0.2541	0.2657	0.2658	4.58%
fireweapon, military	fireweapon	0.0602	0.0622	0.0628	4.33%
waterbody, waterfall	waterbody	0.1317	0.1369	0.1369	3.93%
anchor, studio	anchor	0.6192	0.6358	0.6358	2.69%
flag, flag_usa	flag	0.1196	0.1224	0.1224	2.35%
vehicle, car	vehicle	0.2706	0.2725	0.2726	0.72%
horse_racing, horse	horse racing	0.0003	0.0003	0.0003	0.71%
vehicle, truck, car	vehicle	0.2706	0.2724	0.2725	0.69%
building, house, tower	building	0.2326	0.2322	0.2341	0.62%
building, tower	building	0.2326	0.2322	0.2341	0.61%
fireweapon, walking_running, military	fireweapon	0.0602	0.0603	0.0605	0.54%
flag_usa, flag	flag usa	0.1568	0.1573	0.1573	0.31%
night fire, explosion	night fire	0.2489	0.2495	0.2495	0.25%
duo_anchor, beach	duo anchor	0.1080	0.1080	0.1083	0.22%
duo_anchor, swimmingpool, beach	duo anchor	0.1080	0.1080	0.1083	0.22%
lahoud, chair	lahoud	0.1151	0.1152	0.1152	0.06%
building, house	building	0.2326	0.2327	0.2327	0.02%
explosion, night fire	explosion	0.0782	0.0782	0.0782	0.01%

explosion, night fire, candle	explosion	0.0782	0.0782	0.0782	0.01%
graphics, maps	graphics	0.3815	0.3815	0.3815	0.01%
face, people	face	0.8921	0.8922	0.8922	0.01%
horse, horse_racing	horse	0.0003	0.0003	0.0003	0.01%
chair, lahoud	chair	0.2613	0.2613	0.2613	0.00%
drawing_cartoon, drawing	drawing cartoon	0.1811	0.1811	0.1811	0.00%
duo_anchor, swimmingpool	duo anchor	0.1080	0.1080	0.1080	0.00%
cycling, bicycle	cycling	0.8875	0.8875	0.8875	0.00%
drawing, drawing_cartoon	drawing	0.0440	0.0440	0.0440	0.00%
cartoon, drawing_cartoon	cartoon	0.2783	0.2783	0.2783	0.00%
explosion, candle	explosion	0.0782	0.0782	0.0782	-0.01%
vehicle, truck	vehicle	0.2706	0.2706	0.2706	-0.01%
drawing_cartoon, drawing, cartoon	drawing cartoon	0.1811	0.1811	0.1811	-0.02%
drawing_cartoon, cartoon	drawing cartoon	0.1811	0.1811	0.1811	-0.02%
walking_running, fireweapon	walking running	0.3379	0.3378	0.3378	-0.02%
studio, anchor	studio	0.6653	0.6645	0.6645	-0.12%
graphics, maps, charts	graphics	0.3815	0.3809	0.3809	-0.16%
graphics, charts	graphics	0.3815	0.3790	0.3790	-0.65%
military, fireweapon	military	0.2370	0.2350	0.2349	-0.88%
fireweapon, walking_running	fireweapon	0.0602	0.0596	0.0596	-1.05%
car, vehicle	cartoon	0.2458	0.2432	0.2432	-1.08%
waterbody, beach	waterbody	0.1317	0.1260	0.1302	-1.17%
candle, explosion	candle	0.0801	0.0784	0.0784	-2.16%
boat, waterbody	boat	0.0834	0.0790	0.0790	-5.27%
truck, vehicle	truck	0.0418	0.0348	0.0348	-16.65%
beach, duo_anchor	beach	0.0652	0.0005	0.0276	-57.70%
beach, waterbody	beach	0.0652	0.0037	0.0082	-87.39%
beach, waterbody, duo_anchor	beach	0.0652	0.0036	0.0076	-88.41%
waterfall, waterbody	waterfall	0.4152	0.0053	0.0053	-98.72%