Complex Query Learning in Semantic Video Search

Jin Yuan

Department of School of Computing

National University of Singapore

A thesis submitted for the degree of

Doctor of Computing

2012

Acknowledgements

This thesis contains my research works done during the last four years in School of Computing, National University of Singapore. The accomplishment in this thesis has been supported by many people. It is now my great pleasure to take this opportunity to thank them.

First and foremost, I would like to show my deepest gratitude to my supervisor, Prof. Tat-Seng Chua, a respectable, responsible and resourceful scholar, who has provided me with academic, professional, and financial support. With his enlightening instruction, impressive kindness and patience, I have made a great progress in my research work as well as English writing and speaking. His keen and vigorous academic observation enlightens me not only in this thesis but also in my future study. I think I could not have a better or friendlier supervisor for my Ph.D career.

I sincerely thank Prof. Xiangdong Zhou. His constructive feedback and comments have helped me to develop the fundamental and essential academic competence. I would also like to thank Dr. Zheng-Jun Zha, Dr. Yan-Tao Zheng and Prof. Meng Wang whom I have collaborated for my Ph.D research. Their conceptual and technical guides have helped to complete and improve my research work. I would also like to extend my thanks to all the members in my lab as well as the whole department. The discussion and cooperation with the lab members have given me many useful and enlightening suggestions for my research work, and the life and financial support from the computing department have provided me material assistance to finish my Ph.D career. I really enjoy the four years of Ph.D life with all my teachers, and friends in Singapore.

Finally, I need to express my deepest gratitude and love to my parents, Guihua Yuan and Guizhen Zhang, for their dedication and the many years of support during my former studies that provided the foundation for my Ph.D work. Without their care and teaching, I can not enjoy my Ph.D life. Also, I would like to thank everybody who was important to my growing years, as well as expressing my apology that I could not have thanked everyone one by one. Thank you.

Abstract

With the exponential growth of video data on the Internet, there is a compelling need for effective video search. Compared to text documents, the mixed multimedia contents carried in videos are harder for computers to understand, due to the well-known "semantic gap" between the computational low-level features and high-level semantics. To better describe video content, a new video search paradigm named "Semantic Video Search" that utilizes primitive concepts like "car", "sky" etc. has been introduced to facilitate video search. Given a user's query, semantic video search returns search results by fusing the individual results from related primitive concepts. This fusion strategy works well for simple queries such as "car", "people and animal", "snow mountain" etc.. However, it is usually ineffective for complex queries like "one person getting out of a vehicle", as they carry semantics far more complex and different from simply aggregating the meanings of their constituent primitive concepts.

To address the complex query learning problem, this thesis proposes a three-step approach to semantic video search: concept detection, automatic semantic video search, and interactive semantic video search. In concept detection, our method proposes a higher-level semantic descriptor named "concept bundles", which integrates multiple primitive concepts as well as the relationship between the concepts, such as "(police, fighting, protestor)", "(lion, hunting, zebra)" etc., to model the visual representation of the complex semantics. As compared to simple aggregation of the meanings of primitive concepts, concept bundles also model the relationship between primitive concepts, thus they are better in explaining complex queries. In automatic semantic

video search, we propose an optimal concept selection strategy to map a query to related primitive concepts and concept bundles by considering their classifier performance and semantic relatedness with respect to the query. This trade-off strategy is effective to search for complex queries as compared to those strategies that only consider one criteria such as the classifier performance or semantic relatedness. In interactive semantic video search, to overcome the sparse relevant sample problem for complex queries, we propose to utilize a third class of video samples named "related samples", in parallel with relevant and irrelevant samples. By mining the visual and temporal relationship between related and relevant samples, our algorithm could accelerate performance improvement of the interactive video search.

To demonstrate the advantages and utilities of our methods, extensive experiments were conducted for each method on two large scale video datasets: a standard academic "TRECVID" video dataset, and a real-world "YouTube" video dataset. We compared each proposed method with state-of-arts methods, as well as offer insights into individual result. The results demonstrate the superiority of our proposed methods as compared to the state-of-arts methods.

In addition, we apply and extend our proposed approaches to a novel video search task named "Memory Recall based Video Search" (MRVS), where a user aims to find the desired video or video segments based on his/her memory. In this task, our system integrates text-based, content-based, and semantic video search approaches to seek the desired video or video segments based on users' memory input. Besides employing the proposed complex query learning approaches such as concept bundle, related samples etc., we also introduce new approaches such as visual query suggestion, sequence-based reranking etc. into our system to enhance the search performance for MRVS. In the experiments, we simulate the real case that a user seeks for the desired video or video segments based on his/her memory recall. The experimental results demonstrate that our system is effective for

MRVS.

Overall, this thesis has taken a major step towards complex query search problem. The significant performance improvement indicates that our approaches can be applied to current video search engines to further enhance the video search performance. In addition, our proposed methods provide new research directions such as memory recall based video search.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background to Semantic Video Search

Recent years have introduced a flourish in user generated contents (UGCs), thanks to the significant advances in mobile device and mobile networking technologies that facilitate the publishing and sharing of contents. In particular, the number of user uploaded videos is increasing at an exponential rate in recent years. According to the statistics from Intel, there are about 30 hours of videos uploaded and 1.3 million video viewers in an internet minute in YouTube [You12], a popular video sharing website. Over the entire Internet, the number of user generated videos is even larger. There are two main reasons for this trend. First, since the mid-1990s, the production and storage of new content as well as the digitization of existing content have become progressively easier and cheaper. Second, video content is more intuitive and efficient than text in expressing situations and physical ideas. As a result, both the number and the volume of user generated videos are growing rapidly.

As an important information carrier, the wealth of videos on the Internet offers a rich resource for users to seek the desired information. For example, a couple would like to find videos about "cooking" to teach themselves how to cook, while a reporter may wish to find interesting video clips about "Iraq war" to support his/her news reports. To meet this demand, modern video search engines such as Google, YouTube and Yahoo! etc have became very popular due

to their ability to help users locate the desired videos according to their queries. However, most of these video search engines provide video search services based only on the textual metadata associated with videos. This "Text-based Video Search" paradigm ([SC96]) may fail when the associated text is absent, incomplete, or unreliable with respect to video semantics. Moreover, a user may want to find just a particular segment inside a video ([KR08]). For example, a lawyer evaluating copyright infringement, or an athlete assessing her performance during the training sessions might be more interested only in specific video segments. Text-based video search engines are difficult to serve these needs.

To complement text-based video search, a new video search paradigm named "Semantic Video Search" [SW09] has emerged in recent years. In this approach, a user's query is first mapped to a few related concepts, and a ranked list of video segments is then generated by fusing the individual search results from related concept classifiers. For example, the query "car on the road" is mapped to the related concepts "car", "road", and "vehicle" etc, and then a ranked list of video segments is returned by fusing the results from these concept classifiers. Compared to text-based video search, semantic video search requires the automatic detection of concepts in videos and does not need any text annotations associated with videos, thus it saves the labeling cost. Moreover, semantic video search is able to provide search results on video segment level, which complements the inadequacy of text-based video search aforementioned. However, this approach is highly depended on the accuracy of concept classifiers, which are generally not of sufficient accuracy for many concepts and queries.

Currently, a great deal of research efforts have been devoted to semantic video search that focus on three aspects: concept detection, automatic semantic video search and interactive semantic video search. In particular, the developed techniques include context-based concept fusion [SN03] and multi-label learning [QHR+07] in concept detection, ontology based [WWLZ08] and data-driven based [JNC09] concept selection methods in automatic semantic video search, adaptive feedback [LZN+08] and concept-segment based feedback [WWLZ08] in interactive semantic video search. Based on these technologies, semantic video search system has achieved some success in providing good search results according to users' queries. As argued in [HYea07], the current semantic video search could

achieve comparable performance as compared to standard text-based video search when several thousand of classifiers with modest performance are available.

1.2 Motivation

Although lots of research efforts have been devoted to semantic video search and have achieved some successes, they mainly focus on simple queries such as "soccer", "car in the road", "snow mountain" and so on. The reason for good performance is that a simple query can be well matched to one or more concepts in semantic video search. However, in real world, users often issue complex queries such as "police fighting protester", "car running in the street at night" and so on. For such complex queries, the performance of semantic video search is usually not satisfactory. This is because a complex query tends to involve complex relationship such as "running" "fighting" and so on. between the concepts in the query, while the simple fusion strategy is usually unable to capture such relationships. Thus, it is an urgent task to improve video search performance for complex query in semantic video search.

Recently, researchers have proposed a variety of approaches to enhance performance of semantic video search in a few aspects such as enhancing concept classifier performance, accurately mapping a query to related concepts, and calculating good fusion weights etc. However, very few research work have attempted to exploit the relationships between concepts in a complex query. This thesis aims to bridge this gap. In addition, we apply and extend the proposed approaches to a real world video search task named "Memory Recall based Video Search" to further verity the effectiveness of our proposed approaches. This application demonstrates that the proposed complex query learning approaches work well in a simulated situation and have promising potential to be incorporated into the real world applications.

1.3 The Basic Components and Notations

Given a user's query which may be textual words and/or image samples, Figure 1.1 shows the video search process in which the search system returns the

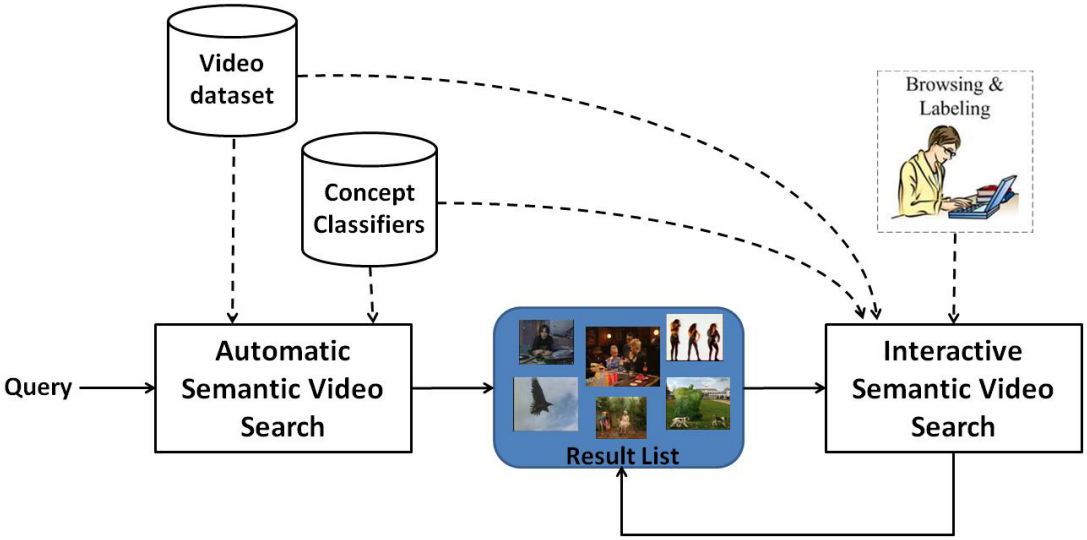


Figure 1.1: The framework of semantic video search system

search results by automatic and interactive semantic video search based on a set of concept classifiers. Generally, the semantic video search is composed of three main parts: Concept Detection [SWG+06a; YCKH07; NS06; JYNH10] which provides a set of concept classifiers to support semantic video search, Automatic Semantic Video Search [CHJ+06; WNJ08] that generates an initial video search results based on users' queries and concept classifiers, and Interactive Semantic Video Search [PACG08; ZNCC09] that involves users' interaction to further refine the search results.

1.3.1 Concept Detection

Concept detection aims to provide a set of concept classifiers to support semantic video search. Figure 1.2 demonstrates the concept detection in two stages: training stage and testing stage. In the training stage, a set of concept classifiers f_k are learned for each pre-defined concept C_k based on its training samples $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where N is the number of the training samples. Here, $\mathbf{x}_i$ is a feature vector extracted from a keyframe, which is a representative frame in a video shot. y_i is the label of $\mathbf{x}_i$ and $y_i = 1$ if the sample $\mathbf{x}_i$ contains the concept C_k, -1 otherwise. In the testing stage, each testing sample is fed to the learned concept classifiers to

generate confidence scores with respect to all the pre-defined concepts. Generally, there are four main steps in concept detection: Video Segmentation, Labeling, Feature Extraction, and Classifier Learning. Here, we elaborate the four steps as below:

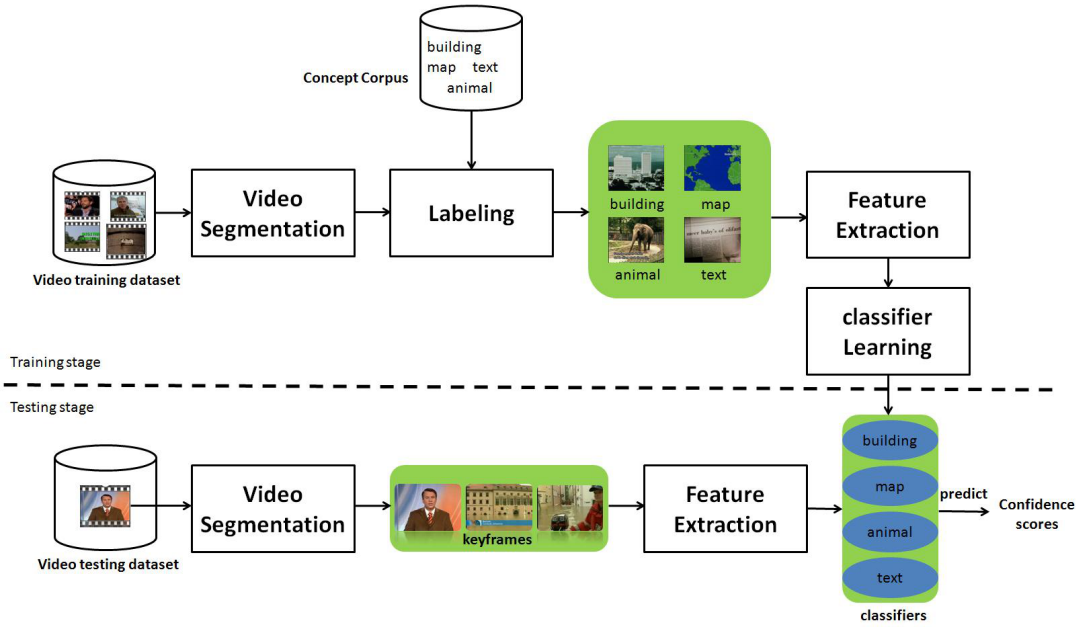


Figure 1.2: An example to illustrate the process of concept detection

- **Video Segmentation**: Video segmentation aims to partition a video into a sequence of video shots. Here, the widely used video segment is the video shot, which is defined in [Han02] as: "a series of interrelated consecutive frames taken contiguously by a single camera shooting and representing a continuous action in time and space". For ease of analysis and computation, a segmented video shot is often represented by a single frame, the so-called "keyframe" [GKS00]. Typically, the central frame of a shot is taken as the keyframe, but many more advanced methods exist such as [BMM99].

- **Labeling**: Based on the extracted video shots in the training set, human users are asked to manually label these shots with respect to each pre-defined semantic concept. For example, if a video shot contains the concepts "car" and "road", then the user should give these two labels to

the shot. To ensure the quality of labeling result, each shot is usually given to several users to label. The final labeling result is generated according to the majority voting scheme.

- **Feature Extraction**: The goal of feature extraction is to derive a compact, yet descriptive representation of the pattern of interest. Typical features to describe a video include text features, audio features, visual features, and their combinations. Since the dominant information in a video is encapsulated in the visual stream, the most common feature is visual feature and it is widely used in many concept detection methods [SWG+06a][YCKH07]. For simplicity, I only focus on visual feature to perform the concept detection task in this study.

- **Classifier Learning**: Given a set of concepts, classifier learning aims to learn a classifier f_k for each concept C_k based on the given training samples $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$. Basic classifier learning approaches include Supervised Learning, such as SVM [CB98], and Semi-supervised Learning such as Graph-based learning [Bis06]. More advanced classifier learning approaches are surveyed in [SW09]. Based on the learned classifier f_k, for each testing sample, the classifier outputs a confidence score to represent the probability of this sample containing the concepts C_k.

1.3.2 Automatic Semantic Video Search

Given a set of concept classifiers $\{f_k\}_{k=1}^{K}$, automatic semantic video search returns an initial result list based on user's queries. Figure 1.3 shows an example. The text query "car at night street" is first mapped to related concepts "car", "night" and "street" by Concept Selection, then the search results are generated by Result Fusion.

- **Concept Selection**: Given a query, concept selection is used to find an appropriate set of concepts to interpret the meanings of query. The widely used concept selection approaches rely on the textual similarity between the query and the concept name [CHJ+06; WLLZ07]. For example, the query "car at night street" is textually similar to the concepts "car", "night" and

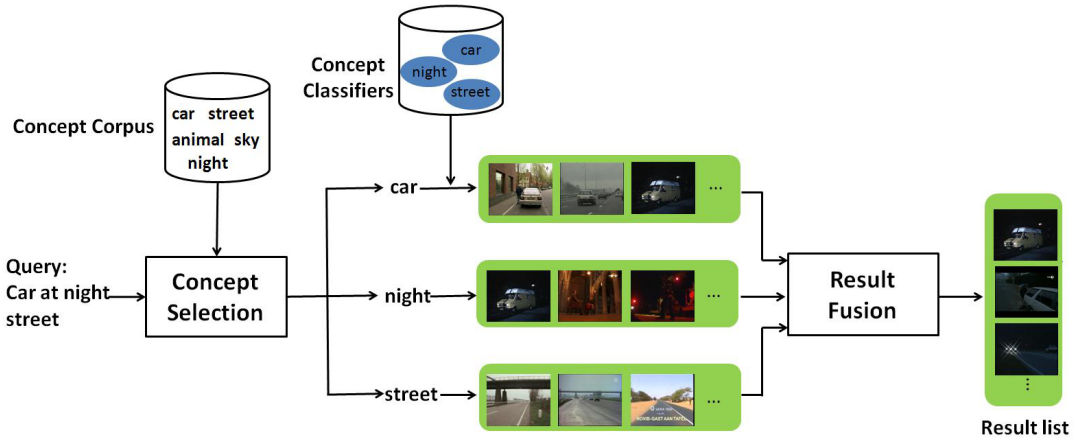


Figure 1.3: An example to illustrate the process of automatic semantic video search

"street", and thus it is mapped to these three concepts. More advanced approaches could explore conceptual correlation to find potentially related concepts [NHT+07]. For example, the query "rabbit" could be mapped to the concept "animal" by using ontology, which models "the-kind-of" relationship between the two concepts.

- **Result Fusion**: Based on the selected related concepts, result fusion integrates search results from these selected concept classifiers. The most popular fusion approach linearly combines search results from these concept classifiers, where the fusion weights are determined according to the importance of each selected concept with respect to the query. For example, the query "person in kitchen" is mapped to two concepts: "person" and "kitchen". Apparently, the concept "kitchen" is more important than "person" since person is too common in videos. Thus, the fusion weight of "kitchen" should be much larger than that of "person". To determine the concept importance, the most popular approach is to employ information retrieval technique to measure text-matching score between concept names and queries [CHJ+06]. The other approaches determine the concept importance according to both text-matching and visual-matching scores [WLLZ07]. More advanced approaches can be found in [SW09].

1.3.3 Interactive Semantic Video Search

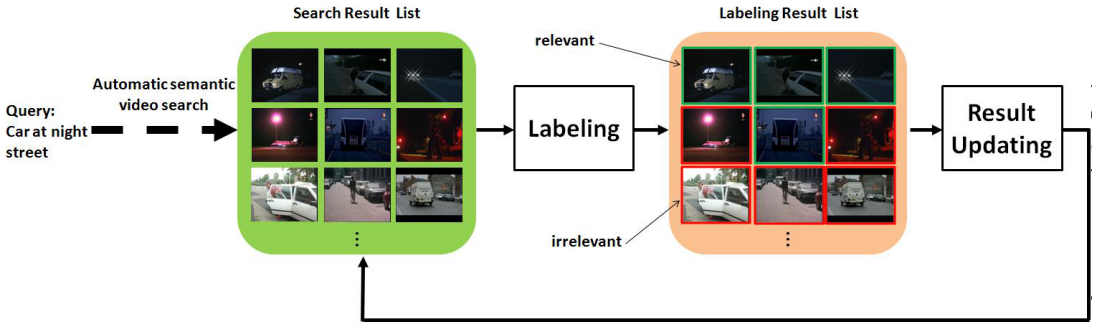


Figure 1.4: An example to illustrate the process of interactive semantic video search

The initial search results from automatic semantic video search may be unsatisfactory. As a result, interactive semantic video search utilizes the interaction between user and system to further refine the search results. Figure 1.4 illustrates the process of the typical interactive search system consisting of two serial steps: Labeling which asks a user to label the search results as relevant or irrelevant, and Result Updating which refines search results based on the new labeled samples. These two steps are repeated until the user is satisfied with the search results.

- **Labeling**: Given a sample list returned by the search system, the user is allowed to label each result sample. Generally, there are two kinds of samples: relevant sample which means that the sample satisfies the query, and irrelevant sample which indicates that the sample does not meet the query.

- **Result Updating**: Based on the new labeled samples, result updating aims to update the search model for a better search result. Generally, the labeled samples especially relevant samples can provide useful information, such as visual information, temporal information, to refine the search results. Specifically, in semantic video search, these labeled samples can be used to adjust the fusion weights to achieve a better fusion result [TRSR09; HLRYC06].

1.4 Complex Query Learning in Semantic Video Search

1.4.1 Definition

In this thesis, we divide queries in semantic video search into two categories:

- **Simple Query**: This category of queries contains one or more co-occurring semantic concepts without specific relationships between the concepts. Examples of this category are "car", "car on the road", "snow mountain" and so on.

- **Complex Query**: This category of queries contains at least two semantic concepts with a specific relationships between them. Examples of this category are: "police fighting protestor", " motorcycle racing at night street", "a couple dancing in the wedding" and so on.

While typical fusion strategies in semantic video search can well interpret the meaning of a simple query, it is difficult to reveal and model the relationships between the concepts in a complex query. In this thesis, we aim to tackle the complex query learning problem in semantic video search. In addition, this thesis ignores some extremely complex queries, such as "Find the video shot with a black frame titled "CONOCER Y VIVIR"", "Find the video shots with a man speaking Spanish" etc, which are usually out of the capability of semantic video search. This is because these queries may need extra techniques such as ASR, OCR etc. to reveal the textual information in videos.

1.4.2 Challenges

There are several challenges for learning complex queries in semantic video search:

- First, a complex query carries semantics that are more complex than and different from simply aggregating the meanings of their constituent primitive concepts. Thus the simple aggregation strategies that can only model semantic concept co-occurrence are unable to capture the specific relationships and interactions between the concepts in a complex query.

- Second, it is well known that the output of concept classifiers in concept detection can be unreliable. Therefore, given a complex query, errors from multiple related concept classifiers may affect the fusion result for the semantic video search. For example, for the query "bird on a tree", the search results are generated by fusing the individual results from the ranking lists of concepts "bird" and "tree". An incorrect result from the classifier for"bird" may have a high confidence score and will thus rank high in the fusion result if the semantic video search simply combines the search results from both classifiers.

- Third, a complex query usually has sparser relevant samples as compared to that of simple queries. This sparse relevant sample problem will severely limit the performance improvement in interactive video search since a user may only have few or no relevant samples to label in the interactive search process.

1.4.3 Overview of the Proposed Approach

To tackle the problems discussed above, in this thesis, we propose three approaches for concept detection, automatic semantic video search, and interactive semantic video search. We briefly summarize the approaches as follows:

- **Concept Bundle Learning**: We propose a higher-level semantic descriptor named "concept bundle", which is a composite semantic concept integrating multiple primitive concepts as well as the relationships between the concepts, such as ("lion", "hunting", "zebra"), ("lady", "laughing", "interview") and so on. Compared to primitive concept, concept bundle carries more complex semantic meanings, and thus it is expected to better meet the video search requirement in a finer granularity. To effectively learn concept bundle, the approach first selects the informative concept bundles, which are measured according to two criteria: users' interest to select those concept bundles frequently used in users' queries, and co-occurrence to select the concept bundles whose constituent primitive concepts tend to co-occur in videos. We use a weight to balance these two criteria. We then learn a

robust classifier for each selected concept bundle under the framework of SVM based multi-task learning.

- **Bundle-based Automatic Semantic Video Search**: Based on the learned classifiers of concept bundles and primitive concepts, the automatic semantic video search needs to select a proper set of concept bundles and primitive concepts to interpret the users' query. For example, the query "person dancing in the wedding" could be directly mapped to the concept bundle ("person", "dance", "wedding"). To accurately select the approximate concepts, we propose a selection strategy that maps the query to related primitive concepts and concept bundles by considering their classifier performance and semantic relatedness with respect to the query. We implement the selection strategy by using a greedy algorithm to save computational cost. The final search results are generated by fusing the individual results from these selected primitive concepts and concept bundles.

- **Related Sample based Interactive Semantic Video Search**: To overcome the sparse relevant sample problem for complex query in interactive video search, we propose a new sample class named "Related Samples". Related samples refer to those video segments that are partially relevant to the query but do not satisfy the entire search criterion. For example, the related samples of the query "car at night street" are the samples that contain the individual concepts "car", "night", or "street" rather than the scene of "car at night street". Generally, related samples are mostly visually similar and temporal neighboring to relevant samples. Moreover, there are much more related samples than relevant ones in the search process. Based the labeled relevant, related and irrelevant samples, we develop a visual-based ranking model, a temporal-based ranking model, as well as an adaptive fusion method to update search results.

To illustrate the advantages of our proposed approaches above, we compare our approaches with the state-of-art methods in three aspects: concept detection, automatic semantic video search, and interactive semantic video search. The key difference between our approaches and that of the existing state-of-art methods are illustrated in Table 1.1.

Table 1.1: The illustration of the differences between our approaches and the existing methods on concept detection, automatic semantic video search, and interactive semantic video search

	The existing work	Our approaches
Concept Detection	1: Focus on learning primitive concept [YCKH07; MZLea08; CHJ+06]. 2: Utilize the relationship between primitive concepts to enhance performance [NKH02; WTS04; QHR+07]. 3: Primitive concepts cannot capture complex query well.	1:Focus on learning concept bundle. 2: Explore the relationship between concept bundle and its primitive concepts to effectively learn classifiers. 3: Concept bundle is semantically closer to complex query.
Automatic Semantic Video Search	1: A query is mapped to primitive concepts. 2: The errors in the fusion result may come from multiple related primitive concept classifiers. 3: The concept selection relies on the concept importance [CHJ+06], or both concept importance and classifier performance with a manual balancing weight [NZKC06].	1: A query is mapped into primitive concepts and concept bundles. 2: The errors in the fusion result may only come from a related concept bundle. 3: An optimization algorithm is devised for concept selection by balancing concept importance and classifier performance.
Interactive Semantic Video Search	1: Work on relevant and irrelevant samples only. 2: May suffer from the sparse relevant sample problem for complex query.	1: Work on relevant, related and irrelevant samples. 2: Alleviate the sparse relevant sample problem by labeling related samples.

Finally, we summarize the contributions of our approaches as follows:

- In chapter 4, we moved a step ahead by proposing a high-level semantic descriptor named "Concept Bundle" to interpret complex query more precisely. The proposed concept bundle selection criterion could effectively find some useful concept bundles so as to reduce the number of concept bundles to be learned. Moreover, the proposed multi-task SVM algorithm can well learn the classifiers for the concept bundles, which could achieve at least 10% improvement in performance as compared to the state-of-art approaches.

- In chapter 5, we developed a concept selection strategy to map a query into related primitive concepts and concept bundles. The greedy algorithm is used to implement this strategy to save the computational cost. In the experiments, we discover that the use of concept bundle is effective to enhance the search performance, and the use of our concept selection strategy could achieve better search performance as compared to the state-of-art approaches in TRECVID 2008 search task.

- In chapter 6, we proposed the idea of related samples to overcome the sparse relevant sample problem for interactive video search with complex query. We employ incremental learning technique to ensure near real-time interactive video search. The experimental results demonstrated that the use of related samples are effective to enhance the interactive search performance for complex queries, and our proposed approach achieves at least 90% performance improvement as compared to the state-of-art approaches.

1.5 Application: Memory Recall based Video Search

To further validate the effectiveness of the proposed complex query learning approaches in semantic video search, we apply and extend the proposed approaches to a real world video search task named "Memory Recall based Video Search" (MRVS). In this task, a user wishes to find a desired video or video segments

that he/she has seen before based on his/her memory recall. A user may input a combination of text description, visual examples and/or concepts to demonstrate the scene in his/her memory. The text description is used to express the textual information about the desired video, while the visual and concept queries are used to depict the visual scenes in his/her memory on the desired video segments. To this end, we develop a multi-modality based video search system to find the desired video or video segments for users. We choose to apply our complex query learning approaches in MRVS task for two reasons: First, in MRVS task, visual scenes in a user's memory usually carry more complex semantic and concept information as compared to the pure text-based complex queries. Therefore, our proposed concept bundles are naturally more effective in this task; Second, the desired video or video segments are unique for each query in MRVS task, which leads to the problem of extremely sparse relevant sample. Our proposed interactive video search technique is able to handle this sparse relevant sample problem with the proposed use of related samples.

1.6 Outline

The rest of this thesis is organized as follows:

Chapter 2 describes works related to this thesis. We first review related work in semantic video search from concept detection techniques, automatic semantic video search and interactive video search. Next, we briefly introduce related work on the other video search approaches including text-based video search, content-based video search, and multi-modality based video search.

Chapter 3 gives an overview of the datasets to be used in this thesis.

Chapter 4 presents the concept bundle learning approach, which is composed of two parts: the informative concept bundle selection, which selects informative concept bundle based on its frequency on the suggested queries by Web video search engine and the concept co-occurrence in the tags of Web videos, and the classifier learning algorithm, which jointly learns all the classifiers of a concept bundle and its primitive concepts by an SVM based multi-task learning.

Chapter 5 introduces the bundle-based automatic semantic video search approach. In this approach, we focus on selecting related primitive concepts and

concept bundles to interpret a user's query. An optimization algorithm is devised to map a query to related primitive concepts and concept bundles by considering their classifier performance and semantic relatedness with respect to the query.

Chapter 6 elaborates related sample based interactive semantic video search. A new sample class named "related sample" is proposed to overcome the sparse relevant sample problem. To effectively explore the labeled relevant, related and irrelevant samples, we propose a visual-based ranking model and a temporal-based ranking model. Moreover, an adaptive fusion method is devised to further boost the search performance.

Chapter 7 applies and extends our proposed approaches to a "Memory recall based Video Search" task, where a multi-modality based video search system is employed to search for the specific scenes according to a user's memory.

Finally, Chapter 8 draws conclusions of our thesis and proposes future work.

Chapter 2

Literature Review

In this chapter, we mainly review related work in semantic video search which is more related to this thesis. After that, we briefly introduce other video search approaches including text-based video search, content-based video search and multi-modality based video search.

2.1 Semantic Video Search

We introduce semantic video search from its three steps: concept detection, automatic semantic video search and interactive semantic video search.

2.1.1 Concept Detection

Early researches aimed to yield a variety of dedicated methods exploiting simple handcrafted decision rules to map restricted sets of low-level visual features, such as color, texture, or shapes, to a single high-level concept. Typical methods are news anchor person in [ZTSG95], sunsets in [SC97], indoor and outdoor in [SP98]. However, such dedicated approaches are limited when many concepts are needed to be detected for semantic video search. As an adequate alternative for dedicated methods, generic approaches for large-scale concept detection have emerged [ABC+03; NH01; SWG+06b]. For example, the MediaMill-101 in [SWG+06a] utilized a corpus of 101 concepts for semantic video search, while Columbia374 in [YCKH07] and VIREO-374 in [JYNH10] leveraged a larger set

of 374 concept classifiers for video search. These 374 concepts are a subset of LSCOM, which is a concept ontology for multimedia search consisting of more than 2000 concepts [NS06].

The generic concept detection approaches typically comprise three steps: video segmentation, feature extraction and classifier learning. Video segmentation divides a video sequence into a set of segments. The most natural candidate for this segment is called "video shot" [DSP91; GKS00]. For each extracted video segment, feature extraction algorithms aim to extract various features, such as text feature [MRS09], visual feature [GBSG01; GS99] and audio feature [Lu01]. In this work, we use visual features like color [GBSG01; GS99], texture [JF91; MM96], and shape [LLE00; VH01] due to their popularity and effectiveness. Based on the extracted features, classifier learning aims to learn a robust concept classifier for each semantic concept. The classical classifier learning approaches include supervised learning [JYNH10; SWG+06a; TLea08] and semi-supervised learning [WHHea09; JCJL08]. The most representative algorithms for these two learning schemes are support vector machine (SVM) [CB98] and graph-based learning [Zhu05]. Next, we will provide more details in these two approaches as well as the classical variants of applying the two methods on concept detection task.

2.1.1.1 Supervised Learning

Suppose that there is a classifier $\mathbf{Y} = f(\mathbf{X}) + \varepsilon$, where $\mathbf{X}$ is the observed input values, and $\mathbf{Y}$ is the output values by the classifier, supervised learning attempts to learn f by observed samples through a learning algorithm. One observes the system under study assembles a training set of observations $\tau = (\mathbf{x_i}, y_i), i = 1, ..., N$, where N is the number of training samples. The observed input values to the system $\mathbf{x_i}$ are fed into a learning algorithm, which produces outputs $f(\mathbf{x_i})$ in response to the inputs. Generally, the learning algorithms attempt to make f bridge the difference between the generated value $f(\mathbf{x_i})$ and the true value y_i. However, this usually leads to the over-fitting problem [JDM00], where the learned classifiers only have a good performance on training set instead of testing set. As a result, the typical supervised algorithms add a generalization term to avoid the over-fitting problem.

Next, we first introduce the classic supervised learning algorithm Support Vector Machine [CB98], then we review the related work on fusion strategies which could improve the classification result. Finally, more advanced supervised learning approaches designed for concept detection are introduced.

Support Vector Machine

In concept detection, Support Vector Machine (SVM) [CB98] is the most popular supervised learning algorithm. It learns a decision hyperplane to separate an n-dimensional feature space into two different classes: one class representing the concept to be detected and one class representing all other concepts, or more formally $y_i = \pm 1$. A hyperplane is considered optimal when the distance to the closest training examples is maximized for both classes. This distance is called the margin. To learn the optimal hyperplane, the objective function of SVM contains two parts: a generalization part to avoid the over-fitting problem, and a penalty part to reduce the training errors, which is expressed as follows:

$$\begin{aligned} & \min \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. \qquad & y_i(\mathbf{w}\mathbf{x_i} + b) \geq 1 - \xi_i \qquad i = 1 \ldots N \end{aligned} \tag{2.1}$$

where $f = \mathbf{w}\mathbf{x_i} + b$ is the hyperplane function, C is a parameter that allows to balance training error and model complexity, and ξ_i is a slack variables that are introduced when data is not perfectly separable. One notable advantage of SVM is the introduction of kernel function, as it can map the distance between feature vectors into a higher dimensional space in which the hyperplane separator and its support vectors are obtained.

Once an SVM-based concept classifier is learned, it is necessary to transfer the output of the classifier into a comparable, normalized score so that the concept detection is able to integrate results from multiple information sources (such as visual, text and audio) by different learning models. The most common normalization, which is also used in the search models in this thesis, is the use of a probabilistic measure for the class membership. In [Pla00], the authors proposed that the posterior probability of the concept occurrence C_k follows a sigmoid function of the output score $f(\mathbf{x_i})$ of the sample $\mathbf{x_i}$. This proposition is widely used among researchers. The discriminative model of Platt's posterior probability is

defined as:

$$P(C_k|\mathbf{x_i}) = \frac{1}{1+exp(Af(\mathbf{x_i})+B)} \tag{2.2}$$

where the parameters A and B of the sigmoid function are fitted to the confidence scores of the training collection.

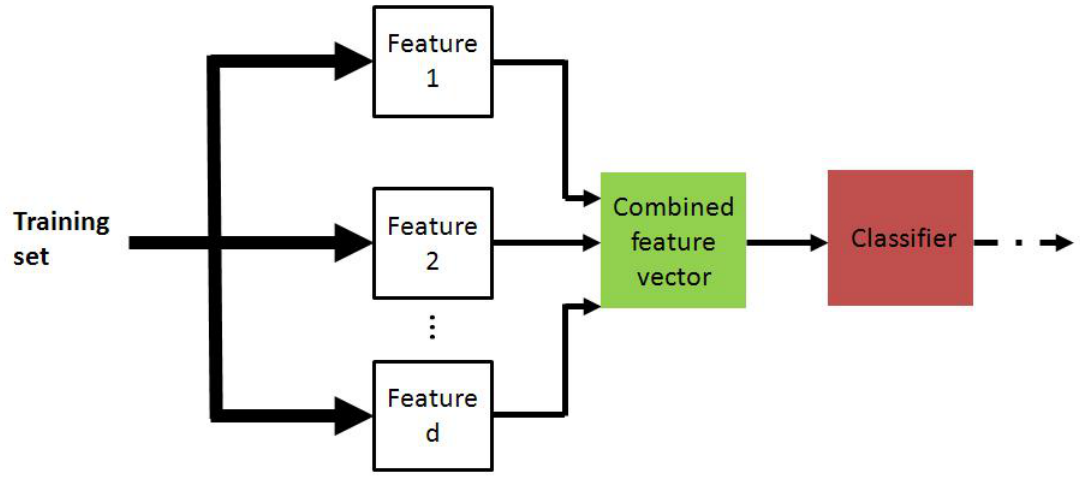


Figure 2.1: General scheme for feature fusion. Output of included features is combined into a common feature representation before a concept classifier is learned.

Fusion

For each visual concept, classifier learning algorithm could generate multiple classifiers based on a variety of features. Thus, a natural question is how to build a general classifier for each visual concept. A typical solution for this problem is to employ fusion strategy which can be divided into two categories: "Feature Fusion" [TLea03; SvGGea06] which first concatenates all kinds of features as a feature vector before learning a classifier, and "Classifier Fusion" [LH02; YCKH07; KHDM98] which individually learns classifiers based on each kind of feature and then fuses the results from individual classifiers.

- **Feature Fusion**: This approach concatenates all kinds of features as a feature vector which is then fed to a classifier learning algorithm to generate a final classifier (see Figure 2.1). For example, Tseng et al. [TLea03] extracted a varied number of visual features, including color histograms, edge orientation histograms, wavelet textures, and motion vector histograms at

both globe and local level. They then simply concatenated all the feature vectors as an aggregated one to learn a concept classifier for each visual concept by SVM. Snoek et al. [SvGGea06] covered more visual features at global, local and keypoint levels to perform concept detection. Besides exploring visual features, some researches [SWH06; HBCea03] import textual features, auditory features, or a combination of both to further enhance the accuracy of concept detection. Although the way of feature fusion is simple and only needs one time learning phrase, it also suffers from the following problems: 1. The concatenation of features will significantly increase the classifier training time; 2. Combining features in an effective way might be problematic, as features often have different layout schemes and numerical ranges. Therefore, in most cases, researchers tend to employ classifier fusion in concept detection.

- **Classifier Fusion**: This approach separately utilizes individual features to train multiple classifiers, which are then combined to generate the final concept detection result (see Figure 2.2). For example, Columbia374 in [YCKH07] individually learned classifiers based on three kinds of visual features (edged direction histogram, Gabor, and grid color moment), and the final concept detection result was generated by averaging the scores from these classifiers. The authors in [LH02] proposed to learn two classifiers by SVM based on visual and textual features respectively. The final concept detection result was generated by averaging results from both classifiers. Besides average fusion, Tseng et al. [TLea03] employed the other five combination operators: (1) minimum, (2) maximum, (3) product, (4) inverse entropy, and (5) inverse variance. In their approach, one of these combination operators was selected to generate fusion result in term of its performance on a validation dataset. The fusion approaches discussed above do not consider the correlation between concept classifiers. As a result, Wu et al. [WCCS04] proposed an optimal multimodal fusion approach. For each concept, the approach first generated several classifiers based on one kind of feature. Then all the training samples were passed to the classifiers to generate a confidence matrix, where the (i, j) element represents the proba-

bility of the sample i satisfying the concept based on the feature j. Finally, this matrix was taken as a new feature matrix to retrain a super-classifier for the concept. This optimal fusion considers the relationship between classifiers, thus obtains a better fusion result. Strat et al. [SBB+12] argued that fusing similar classifiers for a concept is useless, thus they proposed a hierarchical classifier later fusion approach. This approach started with classifier clustering stage, continued with an intra-cluster fusion, and ended with an inter-cluster fusion. Compared to feature fusion strategy, classifier fusion is more efficient and accurate [KHDM98]. Moreover, it is flexible for users to increase efficiency by using simple classifiers for relatively easy concepts, and using more advanced schemes, covering more features and classifiers, for difficult concepts.

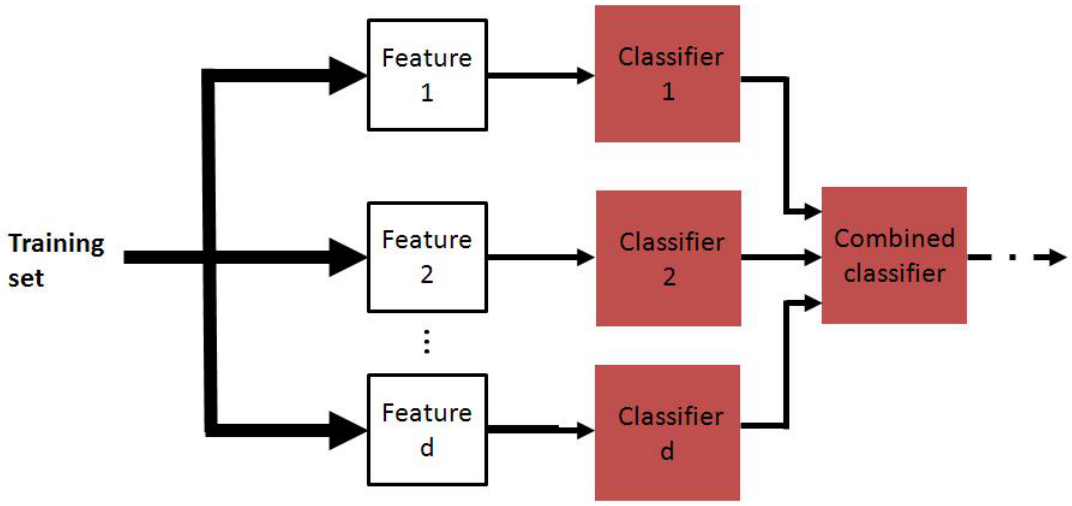


Figure 2.2: General scheme for classifier fusion. Output of feature extraction is used to learn separate probabilities for a single concept. After combination a final probability is obtained for the concept.

Advanced Approaches

The traditional approaches individually and independently learn concept classifier without considering conceptual correlations. Recently, researchers discovered that the conceptual correlations could be explored to enhance the performance in concept detection. For example, once we detect a shot with a certain probability to contain the concept "car", while it is also detected to contain the

concept "road" with a certain probability, we might need to increase probabilities of containing both "car" and "road". To explore conceptual correlations, one well-known approach is to refine the detection results of the individual classifiers with a Context Based Concept Fusion (CBCF) strategy. For example, Wu et al. [WTS04] used an ontology-based multi-classification learning for video concept detection. Each concept was first independently modeled by a classifier, and then a predefined ontology hierarchy was investigated to improve the detection accuracy of the individual classifiers. Smith et al. [SN03] presented a two-step Discriminative Model Fusion (DMF) approach. The approach first constructed model vectors based on detection scores of individual classifiers. Then an SVM classifier was trained to refine the detection results of the individual classifiers. Although the CBCF strategy could enhance the performance, it also suffers from the error propagation problem. This is because the output of the individual classifiers can be unreliable and therefore their detection errors can propagate to the second fusion step. To solve this problem, Qi et al. [QHR+07] proposed a Correlative Multi-label (CML) framework. In this approach, concept classifiers and concept correlations are simultaneously considered in a single step to avoid the error propagation problem. The experimental results demonstrated that this approach achieved better performance than the CBCF approaches.

Besides exploring conceptual correlations, researchers also investigate large-scale video concept detection. For example, Borth et al. [BUB12] proposed how to expand concept vocabularies with trending topics. Their approach first utilize other media like Wikipedia or Twitter to find new interesting topics arising in media and society. Then SVM was employed to learn the classifiers for the new concepts. Geng et al. [GLT+12] proposed the parallel lasso to effectively build robust concept classifiers on large-scale datasets, where Lasso [Tib96] is a sparse learning method designed to tackle high-dimensional feature space by simultaneously performing the sparse feature selection and the model learning. The authors also proposed Parallel lasso to leverage distributed computing to speed up the process of concept classifier learning.

2.1.1.2 Semi-Supervised Learning

In concept detection, the high performance of supervised learning needs a large number of labeled samples, which is limited especially for a large-scale data collection. As a result, researchers turned attention to semi-supervised learning [CZC06]. By leveraging unlabeled data with certain assumptions, semi-supervised learning methods are expected to build more accurate models than those that can be achieved by purely supervised learning methods. Many different semi-supervised learning algorithms, such as self-training, co-training [CZC06], and graph-based methods [Zhu05], have been proposed. Among those approaches, graph-based approach is most popular in concept detection. Next, we first introduce graph-based semi-supervised approach, and then review advanced work by utilizing semi-supervised learning for concept detection.

Graph-based Semi-Supervised Learning

Graph-based semi-supervised learning [Zhu05] performs classification by constructing a graph, where the vertices are labeled and unlabeled samples and the edges reflect the similarities between sample pairs. Let $\mathbf{W}$ be an affinity matrix with W_{ij} indicates the similarity between the i-th and j-th sample. Given two samples x_i and x_j, the similarity W_{ij} is often estimated based on a distance metric $d(.,.)$ and a positive radius parameter σ, i.e.,

$$W_{ij} = exp(-\frac{d(x_i,x_j)}{\sigma}) \tag{2.3}$$

The matrix $\mathbf{W}$ is symmetric. Then a regularization framework is formulated as follows [Zhu05]:

$$f^* = \arg\min_f \left\{ \sum_{i,j} W_{ij} |\frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}}|^2 + \mu \sum_i |f_i - y_i|^2 \right\} \tag{2.4}$$

where $D_{ii} = \sum_j W_{ij}$, and f_i can be regarded as relevance score of x_i. We can classify x_i according to the sign of f_i (positive if $f_i > 0$ and negative otherwise). A noteworthy issue here it how to set y_i. In a binary classification problem, y_i is set to 1 if x_i is labeled as positive, -1 if x_i is labeled as negative, and 0 if x_i is unlabeled.

Let $\mathbf{L} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}$, which is usually named as normalized graph

Laplacian. Eq. (2.4) then has a closed-form solution as

$$f = \left(\mathbf{I} + \tfrac{1}{\mu}\mathbf{L}\right)^{-1}\mathbf{Y} \tag{2.5}$$

where $\mathbf{Y} = [y_1, y_2, \ldots]$ is the initial confidence value set by a user, or predictions by a computer vision model. Alternatively, we can solve the problem sequentially. Applying the update

$$f_{t+1} = \tfrac{1}{1+\mu}(\mathbf{I} - \mathbf{L})f_t + \tfrac{\mu}{1+\mu}\mathbf{Y} \tag{2.6}$$

iteratively results in convergence at f.

Advanced Approaches

In concept detection task, the semi-supervised learning algorithms aim to improve classification accuracy by leveraging both labeled and unlabeled training data. For example, Wang et al. [WHSea06] applied graph model into concept detection based on kernel density estimation approaches. In this approach, each sample (graph node) was represented as a concatenated feature vector from two sources: a color feature vector and an edge feature vector, and then a single graph model was learned for each concept. However, it is noted that feature fusion approach is not effective in concept detection task. Therefore, researchers move their attentions towards classifier fusion approaches. For example, Tong et al. [THL+05] proposed a method to deal with two modalities (text modality and visual modality) in graph-based semi-supervised learning scheme. In their approach, they independently learned a graph model based on each kind of modality, and the final results were generated by fusing the results from both graphs. Wang et al. [WHHea09] extended this method to an arbitrary number of graphs, where they focused on how to determine optimal fusion weights.

The independent concept detection approaches above do not consider the inter-concept relationship [WHSea06; THL+05]. In fact, concepts usually do not occur in isolation (e.g., smoke and explosion). Therefore, more research attentions have been paid to improve annotation accuracy by learning from semantic context. For example, Weng et al. [WC08] utilized contextual correlation and temporal dependencies to improve detection accuracy. In their approach,

they first learned inter-concept correlation and inter-shot dependencies from the training samples. Then they fused the detection results via minimization of the graphical model's potential function, which simultaneously encodes compatibility to classifier's prediction, contextual compatibility and temporal compatibility among nodes. However, in this approach, the learning of contextual correlation is conducted in an offline manner based on training data, resulting in the classical overfitting problem. As a result, Jiang et al. [JWCN09] proposed domain adaptive semantic diffusion to construct an undirected and weighted graph, namely semantic graph, to model the concept affinities. The graph was applied to refine concept annotation results using a function level diffusion process, which simultaneously optimizes the annotation results and adapt the geometry of the semantic graph according to the test data distribution. Alternatively, Weng et al. [WC12] utilized pseudo relevance feedback [YHJ03] to assign unseen testing data pseudo labels based on the initial detection scores. Their approach assumes that a substantial number of top-returned shots from the imperfect detectors are positive and others are negative. Thus, significant patterns found within these temporary labels will likely improve performance.

Although the above approaches achieved some success in enhancing the accuracy of concept detection, they are limited to a small, and fixed concept vocabulary. Recently, researchers began to perform concept detection by exploring online media databases such as YouTube, Flickr. For example, Moxley et al. [MMM98] proposed an approach to automatically annotate multimedia documents based on mining similar documents from online media databases. In their approach, a graph reinforcement method driven by a particular modality (e.g., visual) was used to determine the contribution of a similar document to the annotation target. Then the graph supplies possible annotations of a different modality (e.g., text) that can be mined for annotations of the target.

2.1.1.3 Summary

We summarize related work on concept detection in Table 2.1. Despite the effectiveness of existing works in modeling a single semantic concept, they can not be directly adopted for learning a concept bundle classifier, which integrates multiple

Table 2.1: A summary of the existing related work on concept detection

	independently learn concept classifier	learn concept classifier by exploring conceptual correlation
Supervised Learning	[YCKH07], [TLea03], [SvGGea06]	[WTS04], [SN03], [QHR+07]
Semi-supervised Learning	[WHSea06], [THL+05], [WHHea09]	[WC08], [JWCN09]

semantic concepts and the relationship between the concepts. In particular, the supervised learning methods [JYNH10] suffered from the sparse relevant sample problem, since the relevant samples of concept bundles are usually scare. The semi-supervised learning approaches [WHHea09] explored unlabeled samples to overcome the insufficient training sample problem, but they neglected the relations between the primitive concepts in a bundle. These relations are usually useful in learning the concept bundles. Multi-label learning methods [QHR+07] jointly learned the primitive concepts by exploiting inter-concept relations. However, they focus on boosting the performance for primitive concepts instead of concept bundles.

In Chapter 4, we develop an effective approach for learning concept bundles. Our approach is based on the framework of multi-task learning [AZ05; EP04], which has been applied in wide areas such as object categorization [YY10], multi-device indoor localization [ZPYP08] etc. The basic idea of multi-task learning is to infer an important common structure by simultaneously exploiting the training data from the multiple classifiers which share the common structure. Compared to the previous approaches [AZ05; EP04], our proposed multi-task learning algorithm learns the important common structure (the classifier of the concept bundle) by exploiting the training samples not only from the multiple classifiers (the primitive concept classifiers), but also from the training samples from the common structure.

2.1.2 Automatic Semantic Video Search

In this section, we review related work in automatic semantic video search including research efforts on Concept Selection (Section 2.1.2.1), and Result Fusion (Section 2.1.2.2).

2.1.2.1 Concept Selection

For a given query, concept selection aims to select a set of related concepts to interpret a user's query. According to the query category, we divide the concept selection approaches into two categories: Text-based Concept Selection, and Image-based Concept Selection. Next, we review related work based on this division.

Text-based Concept Selection

- **Term Matching Based Methods**: Given a text query, an intuitive concept selection approach relies on exact text matching between query and concept name. For example, Chang et al. [CHJ+06] proposed a method to select the concepts which are directly mentioned in the query. One shortage of this method is that it ignores many useful concepts that are not directly matched but semantically similar to the query. Furthermore, this approach does not include a component to estimate concept importance.

- **Rule Based Methods**: Term matching based methods cannot capture some implicit correlations between concept and query. For example, the query "president" is difficult to be mapped to the concept "person". As a result, researchers proposed rule based method, which requires users to manually build rules to map a query into related concepts. For example, Natsev et al. [NHT+07] proposed a rule based method which statically maps query terms to concepts using rules. Compared to term matching based methods, rule based method can find some implicit related concepts according to the rules. However, this approach requires users to know all the concepts in advance. Moreover, the production of rules needs enormous manual cost.

- **Ontology Based Methods**: As the rule based methods put heavy burden on users, it is natural for researchers to develop automatic approaches to conduct concept selection. One of the choices relies on ontology such as the WordNet [Fel98], to find useful concepts (see [HAH07; HNN06; SHHe07]). In this approach, WordNet provides a graph to connect all the concepts, and a specific query is mapped to related concepts according to the semantic distance between the query and each concept. There are many semantic distance functions proposed (see [HAH07]). For example, Wu et al. [WP94] defined the semantic distance between two concepts C_i and C_j as follows:

$$WUP(c_i, C_j) = \frac{2D(p_{ij})}{L(C_i,C_j)2D(p_{ij})} \tag{2.7}$$

  where p_{ij} is the common ancestor of the two concepts, $D(.)$ is the depth in the WordNet hierarchy and $L(.)$ is the length of the path between the two concepts.

- **Data-driven Based Methods**: One drawback of ontology based methods is that they only capture the semantic links between concepts. In many cases, the co-occurrence between concepts could be employed in concept selection task. For example, the query "car" can be mapped to the concept "road" since they co-occur with a high probability. To capture the co-occurrence relationship between concepts, researchers proposed data-driven based methods. In this category of approaches, each semantic concept is associated with a concept description from external text source, and the concept selection is performed by measuring the similarity between the query and concept descriptions. For example, Snoek et al. [SHHe07] used the vector space model from [SWY75] to rank concepts for a text query, where a concept description is a short text words of a concept. Hauff et al. [HAH07] also used text retrieval but with a collection of longer concept descriptions which comes from two sources: Wikipedia articles to capture the co-occurrence information between concepts, and WordNet to find the semantic links between concepts. Neo et al. [NZKC06] expanded the query words using internet news articles for better interpretation of query semantics. The expanded query words are then used for classifier selec-

tion, either by direct text matching or the ontology-based selection. One advantage of this approach is that it considers both semantic relatedness and co-occurrence between concepts. To measure the co-occurrence more accurately, Jiang et al. [JNC09] employed Flickr context to reflect the co-occurrence statistics of words in image context rather than textual corpus. They proposed to estimate query-concept similarity by exploiting the context information associated with Flickr images.

Image-based Concept Selection Concept selection can also be done by using visual queries like images. In some cases, image-based concept selection approach is a valuable strategy since it can mine some useful concepts which are not able to be obtained by the text-based concept selection approaches. These useful concepts may be data-adaptive, and thus they are not able to be found by the text-based concept selection approaches. For example, in [LWLZ07], the query "Helicopters in flight" can be mapped to the concept "Mosques" by image-based concept selection approach since the concepts "helicopters" and "mosques" have a high co-occurrence in the training dataset.

Given an image query q, image-based concept selection first calculates the probability $P(C_k|q)$ for each concept C_k. Here, $P(C_k|q)$ represents the probability of the concept C_k occurs in the query image q. Then the top concepts with the highest probabilities are usually selected [SN03]. However, as argued in [SHHe07], selecting the most confident concepts may suffer from the following problems: 1. The most confident concepts are often the least informative ones such as the concepts "person", "outdoor"; 2. The noisy concepts may be introduced. For example, if a user presents an example image where "President Obama" is shown in the desert, the concept "desert" can have a high probability. However, the concept "desert" is not the intention of the user. Hence, it is better to avoid frequently occurring but non-discriminative concepts and favor less frequent but more discriminative concepts. To achieve this goal, Li et al. [LWLZ07] proposed a modified tf-idf method named "c-tf-idf" to estimate relevance between the query and high-level concepts. This method treats each concept as "visual term" and

each image or key frame as "visual document". The c-tf-idf formula is defined as:

$$c - tf - idf(c, d) = freq(c, d)log(\frac{N}{freq(c)}), c \in C \tag{2.8}$$

where C is the concept set, N is the size of corpus, c and d are visual term and visual document respectively, $freq(c, d) \approx P(c|d)$ is the occurrence frequency of c in d, $freq(c) = \sum_d freq(c|d)$ represents the occurrence frequency of c in the corpus. The tf term represents the relevance between a query and a concept, and the idf term measures the popularity of that concept. That means this method selects those relevant and informative concepts for a given query. Alternatively, Natsev et al. [NHT+07] formulated concept detector selection as a discriminative learning problem. Their algorithm mapped the query images to the space spanned by concept detector probabilities. Subsequently, they used support vector machines to learn which concept detectors were most relevant.

2.1.2.2 Result Fusion

Based on the selected concepts, result fusion aims to generate search results by integrating the individual results from these selected concept classifiers. The most popular result fusion approach is linear fusion as shown in Eq. (2.9):

$$r(S_i) = \sum_{k=1}^{K} w_k f_k(\mathbf{x}_i) \tag{2.9}$$

where $f_k(\mathbf{x}_i)$ is the output score of the shot $\mathbf{x}_i$ by the concept classifier of C_k, K is the number of the selected concepts, and w_k is the weight of C_k with respect to the query.

One challenging problem in linear fusion is to determine the fusion weights. Many methods have been proposed to explore how to weight different related concepts. The simplest approach is to set equal weights for the selected concepts, which is widely employed in term matching and rule based concept selection methods [CHJ+06]. However, this average strategy is not reasonable as the selected concepts have different importance to the query. As a result, researchers proposed to determine concept weights according to the importance between concept and

query. For example, Haubold et al. [HNN06] calculated the concept weights according to the semantic closeness between concept and query by WordNet. Snoek et al. [SHHe07] determined the concept weights according to the returned scores by using vector space model between query and concepts. Hauff et al. [HAH07] also used text retrieval technique to calculate concept weights, but considered both semantic closeness and co-occurrence. Besides estimating weights according to text-matching score, Wang et al. [WLLZ07] measured the weights by a linear combination of two scores: text-matching score and visual-matching score. Not only consider the matching scores between concept and query, Neo et al. [NZKC06] argued that concept weight could also be related to the performance of the selected concept classifiers since the use of concept classifier with low performance will introduce noisy results. In their approach, concept weights are set to the product of the classifier accuracy and the similarity between query and concept. Thus the concepts with poor classification performances will be assigned lower weights to avoid noisy results. The above approaches use linear fusion strategy, alternatively, Wang et al. [WWL+08] proposed a query representation approach to capture the concept structure of the query. They gave an example to explain their concept structure as follows: The complex query is "multiple people in formation", and the most salient combination is made up of "Demonstration Or Protest", "Crowd", and "Military". The other combination is made up of "Soldiers", "Crowd", and "Military". This paper proposed a two-level fusion method to solve related concepts combination problem: the bottom level is an AND logic to make sure the selected samples satisfying multiple concepts at the same time, and the upper level uses an OR logic to return the final result from any combination group. Compared to linear fusion, this approach could well capture query structure.

2.1.2.3 Summary

A summary of related work on automatic semantic video search is provided in Table 2.2. Compared to the existing work which selects primitive concepts, in chapter 5, our concept selection approach aims to select primitive concepts and concept bundles to interpret a user's query. Moreover, an optimization algorithm

Table 2.2: A summary of the existing related work on automatic semantic video search from concept selection and result fusion

	Linear Fusion			Non-linear Fusion
	mean weights	weights according to concept importance	weights by considering classifier accuracy	
Term Matching Based Methods	[CHJ+06]			
Rule Based Methods	[NHT+07]			
Ontology Based Methods		[HAH07; HNN06; SHHe07]		
Data-driven Based Methods		[HAH07; SHHe07; JNC09]	[NZKC06], Our approach	[WWL+08]
Image-based Method		[LWLZ07; SN03]		

is devised to perform concept selection by considering semantic closeness and classifier performance between selected bundles and query.

2.1.3 Interactive Semantic Video Search

To further improve search performance, interactive video search [SvdSdR+08; TTR12], which takes users' interaction into consideration, has been proposed. In this section, we review related work for interactive video search from two aspects: Search Technologies and Users' Interface.

2.1.3.1 Search Technologies

The typical interactive search technology is Relevance feedback(RF) [RHOM98] which comprises two steps: (a) the search engine presents the search results and requires users to label them as relevant or irrelevant samples; and (b) the search engine uses the labeled samples to update the search model to improve the search results. To accelerate performance improvement, Active Learning [TC01; HJL06; ZWZ+12], which selects the most informative samples and aims to improve the performance with the least sample size, attracts much attention currently. Active learning attempts to not only optimize the model, but also compute which elements from the unlabeled data pool are the most informative ones. Many typical active learning algorithms have been proposed and found to be effective such as the most ambiguous approach [TC01], angle-diversity [Bri03], error reduction [NA01], and concept-dependent active learning [GCL04] etc.

In the area of video search, Chen et al. [CCHW05] provided two sample selection strategies for users to label: the first one selects the most uncertain samples that are near the SVM hyperplane, and the second simply provides the returned results to users. Goh et al. [GCL04] argued that the active learning faces "scarcity" and "concept isolation" problem in video search task. Here, "scarcity" refers to the rare relevant sample problem in dataset, and "concept isolation" means that the relevant samples are hard to be separated from the irrelevant ones. To solve these two problems, they proposed concept-dependent active-learning (CDAL). This method first selects relevant candidates by keywords to alleviate the scarcity problem. It then employs multiple sampling strategies to select the informative samples from these candidates. Besides employing a single strategy, Luan et al. [LZN+08] argued that different feedback strategies should be adopted at different situations. Therefore, they proposed an adaptive multiple feedback strategies including Recall-driven RF, Precision-driven RF and Locality-driven RF. Those RF strategies will be automatically adopted in search system in terms of an adaptive feedback selection model.

For semantic video search, Christel et al. [CH07] explored concept selection strategies for interactive video search, where they evaluated the performance change of interactive video search under different numbers of selected concepts.

Instead of investigating performance change with respect to the number of selected concepts, Wang et al. [WWLZ08] proposed a feedback strategy at a finer level named "concept-segment", which is defined as a divided district on a probability interval output by a concept classifier. For example, a concept segment may contain all the image samples whose probability values are between 0.1 and 0.2 by the concept classifier. They believed that a good concept-segment is useful for performance improvement. For example, the query "basketball" would be mapped to the high level concept "sports". However, the top-ranked video shots from the concept "sports" may be about the topic of "soccer", while the video shots about "basketball" are hidden in the middle of the "sports" ranking list. Besides investigating the concept selection strategy, researchers also aim to develop effective result updating algorithms in interactive video search process. For example, Pickens et al. [PACG08] represented each video shot as a concept vector where each element represents the presence probability of a concept in this shot. The video shots with a small distance to the relevant shots and large distance to the irrelevant ones were finally returned to the users. Toharia et al. [TRSR09] refined the search results based on concept weight updating during interactive process. In particular, the weights were updated in two ways: manual setting via an interface and automatic adjustment according to the labeled samples. Hauptmann et al. [HLRYC06] transformed concept weight updating to a maximum posteriori probability estimation problem, where the weights were determined by two criteria: minimizing their variation to the previous weights, and maximizing the occurrence likelihood of the labeled samples.

2.1.3.2 User Interface

Besides search technologies, user interface design is also a key factor in interactive video search. A general video search interface displays results in a grid by showing video or video segments one by one. However, this illustration loses some useful information between video shots, such as temporal and content relationship. On the other hand, these information is quite useful for users to find relevant samples in the interactive search process. For example, when a user sees a video shot with "person eating", it is naturally for him to browse the temporal neighboring

Figure 2.3: Cluster-temporal browsing interface ([ROS04]).

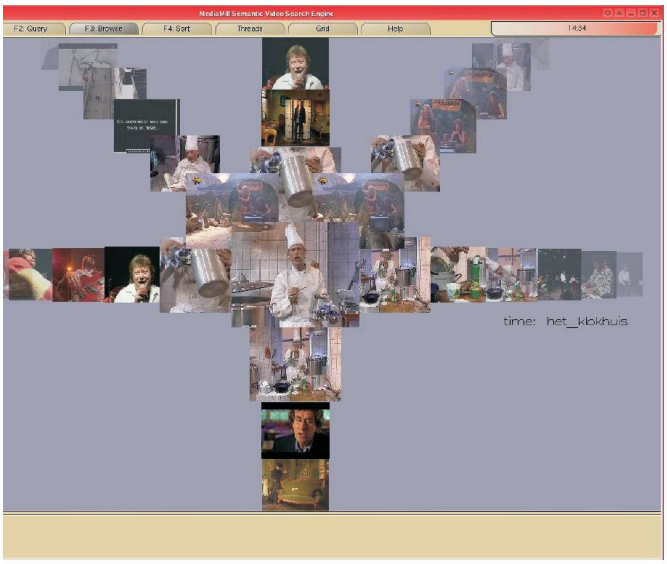


Figure 2.4: The ForkBrowser of the MediaMill semantic video search engine ([dRSW08]).

Figure 2.5: The interface of VisionGo system ([LZN+08]).

Table 2.3: A summary of the existing related work on interactive video search

Sample Selection Strategy	Result Updating Algorithm
Selecting the most uncertain and the most relevant samples [CCHW05]	
concept-dependent active learning to solve the "scarcity" and "concept isolation" problem [GCL04]	
Recall-driven RF, Precision-driven RF and Locality-driven RF with adaptive selection [LZN+08]	
	adjust the weights on different "concept-segments" [WWLZ08]
	adjust concept weights according to manual setting and the calculation on labeled samples [TRSR09]
	adjust concept weights by a maximum posteriori probability estimation on labeled samples [HLRYC06]

samples for the video shots about the topic "kitchen". As a result, researches aim to display multi-model nature of videos on user interface. For example, Rautiainen et al. [ROS04] displayed results model and temporal model in a grid on the interface as shown in Figure 2.3, where the vertical dimension displays the results of content-based retrieval, and the horizontal dimension shows temporal shots. The CrossBrowser presented in [SWKS07] also displayed results in two dimensions: results dimension and temporal dimension, but they do not use a grid in their interface, while Rooij et al. [dRSW07; dRSW08] extended their work by adding more dimensions in the interface including dimensions for visually similar shots, temporal neighboring shots and results ranking list as is shown in Figure 2.4. This visualization allows users to flexibly select a dimension to find relevant samples according to their experience. Besides designing a good visualization, researchers also aim to enhance labeling speed in the interactive search process. For example, Luan et al. [LZN+08] developed a friendly video search interface named "VisionGo". In their system (see Figure. 2.5), user used keyboard to label search results, which has been proved to be more efficient as compared to mouse based labeling.

2.1.3.3 Summary

In this thesis, we focus on developing search technologies for interactive semantic video search, the existing work is summarized from search technology as Table 2.3 shows. Compared to these existing work, in chapter 6, our approach utilizes a new sample class named "related sample" to enhance the interactive semantic video search. Moreover, our approach integrates visual information, temporal information and concept information of the labeled samples to update search results with optimal fusion weights.

2.2 Text-based Video Search

The widely used commercial video search engines such as Google, Bing, YouTube etc. perform video search by exploiting the text annotations associated with the videos. Given a text query, the "Text-based video search" approaches [SC96;

AACea05; CHJ+06] return videos or sometimes video segments by measuring the text similarity between the query and their associated text annotations. For example, Amir et al. utilized the ASR documents to return the video shot results [AACea05], while Chua et al. returned video shot results by exploring both ASR and OCR documents [CNZea06]. Compared to the other video search approaches, the text-based video search are the most widely used video search approach because of its good performance. However, when the text annotations associated with videos are incomplete or inexact, this approach is usually ineffective.

2.3 Content-based Video Search

To complement the text-based video search, "Content-based video search" approaches (CBVR) [GN08; HXLZ11] utilize the low-level visual features such as color, shape, texture etc. to find the relevant videos or video shots based on users' queries. For example, Snoek et al. extracted various visual features including color, texture, SIFT etc. for each video shot, and transformed the features as a codeword vector. They then used the kernel-based learning algorithm [CB98] to return relevant video shots by learning a classifier based on the codeword vectors from the query image and pseudo-negative samples [SvdSdR+08]. Sivic et al. extracted face features from an example shot and matched the extracted features with the stored face features. Then, shots containing the queried face were retrieved [SEZ05]. Besides exploring visual feature, more advanced approaches consider temporal features [WZP00] and spatio-temporal features [YHC04] etc. to enhance the search performance. Although the state-of-art approaches have improved search performance, the performance is still unsatisfactory because of the semantic gap problem [HYea07].

2.4 Multi-modality based Video Search

In the real case, the performance gain of using a single video search approach is usually limited, and thus researchers tend to develop multi-modality based video search systems [CHEea06; KNC05; CHJ+06] that integrate several video search approaches. For example, Kennedy et al. proposed to linearly combine

the search results from text-based and content-based video search approaches [KNC05]. To learn good fusion weights, this approach clustered training queries into different query classes according to search performances. The fusion weights were determined according to the query class of a given query. Chang et al. integrated the search results from text-based, content-based and semantic video search approaches, where they employed the query-class-dependent fusion weights to achieve a high search performance [CHJ+06]. Compared to a single video search approach, multi-modality based video search is more effective in the real-world video search applications.

2.5 Summary

In this thesis, we focus on complex query learning problem in semantic video search. We propose new approaches to enhance the search performance for complex queries in concept detection, automatic semantic video search and interactive semantic video search (see chapter 4, 5, 6). Compared to the existing approaches, the proposed approaches could better tackle the complex query learning problem. To further illustrate the effectiveness of the new approaches, in chapter 7, we apply these approaches in a multi-modality based video search system to tackle a real-world video search task named "Memory Recall based Video Search" (MRVS).

Chapter 3

Overview of Dataset

We evaluate our proposed approaches on two video datasets: the first is the academic video dataset named "TRECVID" dataset, and the second is the Web video dataset called "YouTube" dataset.

3.1 TRECVIDVID Dataset

The TRECVID datasets [TRE08] are provided by the National Institute of Standards and Technology (NIST). In this thesis, we select two TRECVID datasets: TRECVID 2008 dataset (called "TV08" for short), and TRECVID 2010 dataset (called "TV10" for short), to evaluate the search performance of our approaches.

3.1.1 TRECVID 2008 Dataset

The "TV08" dataset consists of three parts (see Table 3.1): a training set that contains 110 videos with 18,120 keyframes from the TRECVID 2007 development dataset, a validation set that contains 109 videos with 18,142 keyframes from the TRECVID 2007 testing dataset, and a testing set that contains 219 videos with 35,766 keyframes from the TRECVID 2008 testing dataset. To represent each keyframe, we extracted three visual features: 166-dimensional color histogram, 100-dimensional edge distribution histogram and 96-dimensional texture cooccurrence. They are concatenated into a 362-dimensional feature vector. In semantic video search, we selected the 374 concept classifiers in [YCKH07] to

Table 3.1: The summary of "TV08" dataset

# of video/keyframe in the training set	# of video/keyframe in the validation set	# of video/keyframe in the testing set
110/18,120	109/18,142	219/35,766

Table 3.2: The summary of "TV10" dataset

# of video/keyframe in the training set	# of video/keyframe in the validation set	# of video/keyframe in the testing set
2,673/100,132	500/19,553	8,471/144,988

perform search on the 48 queries, where the ground truth files were provided by the TRECVID 2008 official dataset.

3.1.2 TRECVID 2010 Dataset

We divided the "TV10" dataset into three parts (see Table 3.2): a training set that contains 2,673 videos with 100,132 keyframes, a validation set that contains 500 videos with 19,553 keyframes, and a testing dataset that contains 8,471 videos with 144,988 keyframes. Each keyframe is represented by two kinds of visual features: 166-dimensional color histogram and 100-dimensional edge distribution histogram. They are concatenated into a 266-dimensional feature vector.

We employed the 130 primitive concepts provided by the TRECVID 2010 and the corresponding concept classifiers were downloaded from [YCKH07]. For automatic video search, we evaluated the search performance on the 298 queries (the queries 8 and 11 were removed officially since they are asked twice) from the TRECVID 2010 KIS task. We evaluated the search performance of interactive video search on the queries 1-24. In this task, a query consists of three parts: a text query, visual queries, and concept queries (see Figure 7.1). The text parts of the 298 queries are provided officially, while the visual and concept queries were provided by 10 users based on their recalls on certain desired videos. We provided a system for users to input a query. The process of generating the visual and concept queries is as follows. We first asked each user to view the relevant video of a query. After 12 hours, the user was asked to search for the same video again by providing visual and concept queries based on his/her memory

Table 3.3: The 41 primitive concepts selected from popular video tags in "YT10" dataset

astronaut	baby	basketball	beach
car	cat	children	cooking
crash	dance	fighting	fire
flood	horse	house	hunting
interview	kitchen	lady	laughing
lion	moon land	motorcycle	mouse
office	person	police	protester
race	riding	river	sea
singing	soccer	street	telephone
television	tree	vehicle	wedding
zebra			

recall. The user can input one, two, or at most three visual and concept queries. For convenience, we implemented the initial system that allows for a maximum of three visual and concept queries. After the user had input an initial visual queries, the system suggested 10 potential visual queries, and the user might select one of the suggested visual queries to replace the original one.

3.2 YouTube Dataset

To evaluate the search performance in the real world, we conducted experiments on the real-world YouTube video datasets. We constructed three YouTube video datasets in 2010, 2011 and 2012 respectively (called "YT10", "YT11" and "YT12" for short). In each year, we expanded the dataset including the number of primitive concepts and that of videos and keyframes.

3.2.1 YouTube 2010 Dataset

To construct "YT10" dataset, we first selected 41 representative primitive concepts from the popular tags in YouTube (see Table 3.3). These concepts cover a wide variety of semantics, including entertainment, arts, communication, sports, and animal etc. We then submitted each concept to YouTube and collected the related queries suggested by the search engine. From the suggested queries, we

Table 3.4: The 20 queries on "YT10" dataset

ID	Query	ID	Query
1	Astronauts carrying out moon landing	11	Lady on the beach with sea visible
2	A person is cooking in kitchen	12	Lions hunting zebras
3	A person is riding a horse	13	Motorcycles racing at night
4	Baby is laughing	14	Persons dancing in the wedding
5	Cars crashing in a race	15	Persons fighting in street
6	Cat and mouse	16	Persons interviewing in street
7	Children are singing	17	Persons playing basketball in street
8	House in flood	18	Police fighting protester
9	Lady laughing in an interview	19	Soccer fighting
10	Lady make a telephone call in office	20	Vehicle fires

Table 3.5: The summary of "YT10" dataset

# of video/keyframe in the training set	# of video/keyframe in the validation set	# of video/keyframe in the testing set
2,002/90,000	990/50,000	357/18,000

selected 20 complex queries as listed in Table 3.4. Finally, we submitted each concept as a query to YouTube video search engine and downloaded the top 80 returned videos. To ensure that each query has some relevant videos, we also submitted each query to YouTube and downloaded about 10 relevant videos. This gives rise to a total of 3,420 videos. After removing the duplicates, we obtained a total of 3,349 videos. For each video, we extracted keyframes and obtained a total of around 158,000 keyframes. This dataset is randomly separated into a training subset containing 2,002 videos with about 90,000 keyframes, a validation subset containing 990 videos with about 50,000 keyframes, and a testing subset including 357 videos with about 18,000 keyframes. For each keyframe, we extracted three visual features: 166-dimensional color histogram, 100-dimensional edge distribution histogram and 96-dimensional texture cooccurrence. They were concatenated into a 362-dimensional feature vector. We summarize "YT10" dataset in Table 3.5.

To obtain the ground truth files for the primitive concepts and queries, we

conducted a manual labeling procedure. Specifically, each keyframe is labeled as a relevant or irrelevant sample with respect to the 41 primitive concepts and the 20 queries. We invited five subjects to manually label the keyframes. Each keyframe is labeled by at least three subjects, and the ground truth is established through majority voting.

3.2.2 YouTube 2011 Dataset

In 2011, we constructed "YT11" dataset by collecting videos from YouTube website. 70 concepts were manually selected from the popular YouTube tags (see Table 3.6). The popularity of these concepts on YouTube implies they are representative of user interest. These concepts cover a variety of domains, including entertainment ("singing"), scene ("lake"), sports ("basketball"), and animal ("horse") etc. We submitted each concept to YouTube search engine and collected the suggested queries. We then selected 40 complex queries for the experiments (see Table 3.7). To construct the video dataset, we issued each concept as a query to YouTube and collected about the top 100 videos. Furthermore, to ensure that each query has relevant videos, we submitted each query to YouTube and downloaded about 10 relevant videos. As a result, we collected $7,662$ videos in total. After keyframe extraction, $314,775$ keyframes were obtained. We randomly divided these videos into three parts: a training set containing $3,370$ videos and $134,613$ keyframes, a validation set containing 500 videos and $24,155$ keyframes, and a testing set consisting of $3,792$ videos and $156,007$ keyframes. To represent the content of each keyframe, we extracted three kinds of visual features: the 166-dimensional color moment, the 100-dimensional edge distribution histogram, and the 96-dimensional texture co-occurrence features, which are widely used and proved to be effective in previous researches [TLea08; CTH+09]. We normalized each dimension of all features, and concatenated them into a single 362-dimensional feature vector. We summarize "YT11" dataset in Table 3.8.

To obtain the ground truth files of each keyframe on the 70 primitive concepts and the 40 queries, we invited ten subjects to manually label the keyframes. Each keyframe is labeled by at least three subjects, and the ground truth is established through majority voting.

Table 3.6: The 70 concepts and their numbers of relevant samples in the training and validation sets of "YT11" dataset

Concept Name (# of relevant videos/keyframes)	Concept Name (# of relevant videos/keyframes)
airplane flying (17/154)	astronaut (14/224)
baby (22/258)	basketball (63/1805)
beach (24/481)	bear (41/841)
bird (20/319)	blackboard (15/221)
bus (12/162)	camel (17/355)
car (57/1268)	cat (44/510)
chair (7/345)	children (14/280)
cityscape (17/272)	classroom (19/585)
computer (7/51)	cooking (57/1822)
crash (7/27)	dance (13/329)
demonstration or protest (44/634)	desert (15/204)
doorway (2/6)	female human face closeup (7/100)
fighting (58/1550)	fire (18/118)
flood (29/563)	hand (5/59)
horse (31/449)	house (18/179)
hunting (41/493)	interview (38/734)
kiss (15/77)	kitchen (34/948)
lady (35/1377)	lake (14/244)
laughing (39/310)	lion (33/647)
moon landing (19/310)	motorcycle (44/1183)
mouse (5/15)	night (5/18)
office (49/1252)	person (61/2245)
person eating (8/117)	person playing a musical instrument (15/81)
person riding a bicycle (24/157)	police (27/229)
protester (16/412)	race (21/914)
riding (22/536)	river (13/101)
santa clause (24/368)	sea (33/649)
ship (34/421)	singing (12/154)
soccer (57/1782)	solar eclipse (2/5)
soldier (18/308)	street (29/652)
swimming (48/692)	tank (9/252)
telephone (22/232)	television (4/69)
traffic intersection (26/140)	tree (38/736)
vehicle (51/1442)	wedding (53/1485)
writing (3/82)	zebra (18/166)

Table 3.7: The 40 queries and their numbers of relevant samples in the testing set of "YT11" dataset

ID	Query (# of relevant videos/keyframes)
1	A bear in a river (11/36)
2	A bird in a tree (17/149)
3	A lady singing (46/650)
4	A lake with trees visible (12/58)
5	A person cooking in kitchen (67/928)
6	A person riding a horse (30/241)
7	A person talking with a telephone in office (32/148)
8	A person writing on blackboard (11/53)
9	A tank with soldiers on it (4/20)
10	Airplane flying (11/146)
11	Astronauts carrying out moon landing (23/146)
12	Baby laughing (17/70)
13	Bus crash (13/71)
14	Camels walking in desert (9/63)
15	Cars crashing in a race (20/191)
16	Cat and mouse (15/74)
17	Chair dance (11/151)
18	Children watching TV (10/27)
19	Classroom fight (8/33)
20	Crazy traffic intersection (17/96)
21	Hand dance (7/134)
22	House in flood (41/427)
24	Motorcycles racing at night (4/26)
23	Lions hunting zebras (8/34)
25	People seeing solar eclipse (5/40)
26	Persons dancing in the wedding (15/261)
27	Person eating (10/167)
28	Persons fighting in a street (20/364)
29	Persons in office with computer visible (34/331)
30	Persons interviewing in street (10/119)
31	Persons kissing in beach (6/16)
32	Person playing a musical instrument (17/179)
33	Persons swimming in sea (18/201)
34	Person riding a bicycle (16/111)
35	Polices fighting protesters (16/159)
36	Santa Claus and children (10/110)
37	Ship crash (6/32)
38	Soccer fighting (17/121)
39	Vehicle fires (12/148)
40	Woman playing basketball (15/216)

Table 3.8: The summary of "YT11" dataset

# of video/keyframe in the training set	# of video/keyframe in the validation set	# of video/keyframe in the testing set
3,370/134,613	500/24,155	3,792/156,007

Table 3.9: The summary of "YT12" dataset

# of video/keyframe in the training set	# of video/keyframe in the validation set	# of video/keyframe in the testing set
4,325/312,032	725/51,130	2,166/146,963

3.2.3 YouTube 2012 Dataset

In 2012, we constructed another dataset named the "YT12" dataset from YouTube web site. To build this dataset, we first selected the 130 primitive concepts from TRECVID 2010 concept detection task [TRE]. We then submitted each concept name as a query to YouTube video search engine and downloaded the videos on the first three pages of search results (about 20/page) together with their titles and tags information (text descriptions). After removing duplicate videos, we obtained 7,216 videos (about 1,000 hours in total). We randomly divided the videos into three sets: a training set that contains 60% videos (4,325 videos with 312,032 keyframes), a validation set that contains 10% videos (725 videos and 51,130 keyframes), and a testing set that contains the remaining videos (2,166 videos with 146,963 keyframes). For each keyframe, we used two kinds of visual features: 166-dimensional color histogram and 100-dimensional edge distribution histogram. They are concatenated into a 266-dimensional feature vector. We summarize "YT12" dataset in Table 3.9.

On this dataset, 50 queries (see the text part of the queries in Table 3.10) were generated with the helps of the 10 users. The process is as follows. We first randomly provided each user with 20 videos from the "YT12" dataset and the user selected an interesting video to view. After a duration of one day (one week, two weeks, two months), the user was asked to find the same video by providing a text query and a sequence of visual and concept queries based on his/her memory recall (see Figure 7.1). Following this process, we generated 20 queries with a duration of one day, 10 queries with a duration of one week, 10

queries with a duration of two weeks, and 10 queries with a duration of two months.

Table 3.10: The 50 queries on "YT12" dataset

Index	Text	Duration
1	a girl wearing a black coat and talking about her work	1 day
2	a man is speaking with a black hat and sitting on a brown chair	1 day
3	a blue sky where a white point like ufo is moving	1 day
4	a cartoon video where a man are complaining about hard working without payment	1 day
5	the song about riding bicycle where sky, cloudy are visible	1 day
6	a news video, where a woman is interviewed to talk about dream ride in the front of a school	1 day
7	the white and black video of several boys in a band is singing	1 day
8	a demonstration where a crowd of people are against off the education cut	1 day
9	two men are fishing in a boat	1 day
10	a man in a boat talk about getting a car into a boat	1 day
11	the video about car racing	1 day
12	the TV videos talking about wetpaint	1 day
13	A women tells us how to do gym	1 day
14	old woman and young man are dancing beside a sea, and a band are singing	1 day
15	a woman is painting a city view on a blank paper	1 day
16	several students getting out of a school bus, and then two boys sitting before computer and playing games	1 day
17	the TV video named movie night to introduce movies	1 day
18	a computer is starting up in a small monitor	1 day
19	a corporate leader talking of sustainability	1 day
20	the video introducing a red car toy	1 day

Continue from Table 3.10

Index	Text	Duration
21	a video about bus traffic in city, persons are interviewed about bus traffic, they select to take subway instead of bus in rush hour	1 week
22	the video talking about asian military training project	1 week
23	a sunrise view, ricefield with flowers, paddy visible	1 week
24	a man going towards a crowd of cows, he is lying on the ground and these cows are watching him	1 week
25	the advertisement where a cow appears in a sea scene, a woman eating yogurt	1 week
26	a beach view, a crowd of people, person swimming	1 week
27	a man and woman talking about flight design for sport	1 week
28	people speaking about building collapse by terrorist attacks	1 week
29	a demonstration in Iran, where a crowd of people are burning and walking in street, the president is speaking	1 week
30	an orange desert view where sky, cloudy are visible	1 week
31	two dogs are sitting on a sofa, then a woman is coming to play the hair of the white dog	2 weeks
32	an ambulance car is running on the road, and the title "passage ambulance" in the final screen	2 weeks
33	the video to illustrate several kinds of cars, including police car, Fire engines cars	2 weeks
34	a girl inside the bus is bouncing from the seat with laughing from the others	2 weeks
35	a car is burning, and firemen are wiping out the fire	2 weeks
36	campus news cast with a girl reading a book behind desk	2 weeks
37	how to use a computer software to create a chair	2 weeks
38	a golf coach indoor is teaching about how to play golf	2 weeks
39	a big dog come out from a room	2 weeks

Continue from Table 3.10

Index	Text	Duration
40	a knife inserting a man's head with blood visible, two man behind him dancing with music in background	2 weeks
41	young demonstrators in street, holding a sign and shouting, polices are trailing on the motorcycles	2 months
42	a crane is slowly raised his arm	2 months
43	a music is playing with Lyrics displaying in the middle of the screen by big font	2 months
44	a man beside a screen displaying a car race	2 months
45	a 3D software to design different poses of persons	2 months
46	Buju banton with many pictures are displaying, a music is playing in background	2 months
47	toy boat in water swimming controlled by the people	2 months
48	two women displaying how to change clincher tire of a bicycle in road	2 months
49	boys and girls are dancing in the beach and singing about beach blanket bingo	2 months
50	a woman is playing a game of shooting	2 months

Chapter 4

Concept Bundle Learning

4.1 Introduction

In semantic video search, the fusion strategy transforms the query semantics into a set of primitive concepts and performs the retrieval by aggregating the search results from different concepts. This approach works well for queries that could be well matched with one or more primitive concepts. However, simple aggregation of results from primitive concepts may fail for video search with complex queries, such as "police fighting protester", "persons fighting in the soccer match" etc.. This is because they carry semantics that are more complex than and different from simply aggregating the meanings of their constituent primitive concepts.

To interpret complex queries well, in this chapter, we move one-step beyond primitive concepts and propose a higher-level semantic descriptor named "concept bundle". A concept bundle is defined as a composite semantic concept that integrates multiple primitive concepts as well as the relationships between the concepts, such as ("lion", "hunting", "zebra") and ("lady", "laughing", "interview") etc.. Carrying higher-level semantics, concept bundle is expected to deliver more precise descriptions of video contents, and thus better meet the video search requirement in a finer granularity.

However, there are two challenges in learning concept bundles:

1. It is intractable to learn all the possible concept bundles for the practically unconstrained set of queries. Hence it is important to select informative

concept bundles to learn.

2. Compared to primitive concepts, the relevant samples for concept bundles are usually scarce. This poses difficulty in deriving robust concept bundle classifiers.

To address these problems, the approach first automatically selects informative concept bundles, with each bundle comprising two or more primitive concepts. We define the informative concept bundles as those frequently used in users' queries and whose constituent primitive concepts tend to co-occur in videos. Thus, the informativeness of a concept bundle can be measured based on its frequency on the suggested queries by Web video search engine and the concept co-occurrence in the tags of Web videos. Second, we learn a robust classifier for each selected concept bundle under the framework of SVM based multi-task learning. Our premise is that since the "concept bundle" is an intersection among its primitive concepts both in terms of semantics and data exemplars, the primitive concept classifiers share a common part in the decision space, which characterizes the firing region of the "concept bundle". We thus formulate each primitive concept classifier as a linear combination of two parts: a private classifier for this primitive concept and a communal classifier for the concept bundle. The concept bundle classifier (i.e., the communal classifier) can be obtained by jointly learning the primitive concept classifiers based on the training samples from both the primitive concepts and the concept bundle. We expect the training samples of the constituent primitive concepts to model the individual concepts that appear in the concept bundle, while that of the concept bundle to model the relationship between the concepts.

We evaluate the proposed approaches on two video datasets: TRECVID 2008 and YouTube 2010 datasets. Compared to the state-of-the-art methods, the MAPs by our concept bundle learning approach achieve at least 19% and 29% improvements on TRECVID 2008 and YouTube 2010 datasets respectively.

4.2 Learning Concept Bundle

In this section, we first propose an approach to select informative concept bundles in Section 4.2.1, then we elaborate our concept bundle learning algorithm in Section 4.2.2.

4.2.1 Informative Concept Bundle Selection

Given a concept corpus $\{C_k\}_{k=1}^{K}$ containing K primitive concepts, it is intractable to learn all the $2^K - K - 1$ possible concept bundles. In this research, we propose to learn the informative concept bundles, which are selected according to two criteria: users' interest and co-occurrence of the involved primitive concepts in video data. The users' interest of a concept bundle can be measured based on the occurrence frequency of this bundle in users' queries, while the co-occurrence of the involved primitive concepts can be inferred from the tags of Web videos. We use external sources from YouTube to help calculate the informativeness of concept bundle. For each individual concept C_k, we submit it to YouTube video search engine as a query and collect the queries suggested by the search engine. These suggested queries are normally related to the concept. We also download the tags of the top ranked videos in the search results. As a result, we obtain a set of suggested queries and a set of tag files, where each tag file records the tags contained in a specific video. Based on these metadata, the informativeness of the concept bundle $(C_1, C_2, \cdots, C_M)$ is defined as

$$I(C_1, \cdots, C_M) = \alpha \frac{N_q(C_1, ...C_M)}{\sum_{m=1}^{M} N_q(C_m)} + (1-\alpha) \frac{N_t(C_1, ...C_M)}{\sum_{m=1}^{M} N_t(C_m)} \tag{4.1}$$

where $N_q(C_1, ...C_M)$ is the number of queries containing the bundle $(C_1, C_2, \cdots, C_M)$, $N_t(C_1, ...C_M)$ is the number of tag files that include $(C_1, C_2, \cdots, C_M)$, and α is a balance weight. The first term represents users' interest on the concept bundle $(C_1, C_2, \cdots, C_M)$ which is measured by its normalized frequency in the suggested queries. The second term measures the co-occurrence of these M concepts in video tags. We combines these two terms with a weight α to measure the informativeness. The value of the informativeness is between [0,1], and the larger

value of $I(C_1, \cdots, C_M)$ means the concept bundle is more informative.

Based on the above measurement, we select the informative bundles whose informativeness exceeds a certain threshold. However, it is computationally heavy to compute the informativeness for all the possible concept bundles. Here we adopt a dynamic programming approach to compute the informativeness in a bottom-up approach. We sequentially compute the informativeness of concept pairs, triples, quadruples, and so on. For any concept m-tuple $(C_1, C_2, \cdots, C_m)$, if its informativeness is less than a threshold, then we can discard any $(m+1)$-tuple that covers $(C_1, C_2, \cdots, C_m)$ since its informativeness will certainly be less than the threshold. This can be proved as follows:

$$
\begin{aligned}
&\because N_q(C_1, ...C_M) \geq N_q(C_1, ...C_M, C_{M+1}), \sum_{m=1}^{M} N_q(C_m) < \sum_{m=1}^{M+1} N_q(C_m) \\
&\Rightarrow \frac{N_q(C_1,...C_M)}{\sum_{m=1}^{M} N_q(C_m)} > \frac{N_q(C_1,...C_M,C_{M+1})}{\sum_{m=1}^{M+1} N_q(C_m)}, \\
&N_t(C_1, ...C_M) \geq N_t(C_1, ...C_M, C_{M+1}), \sum_{m=1}^{M} N_t(C_m) < \sum_{m=1}^{M+1} N_t(C_m) \\
&\Rightarrow \frac{N_t(C_1,...C_M)}{\sum_{m=1}^{M} N_t(C_m)} > \frac{N_t(C_1,...C_M,C_{M+1})}{\sum_{m=1}^{M+1} N_t(C_m)} \\
&\therefore I(C_1, \cdots, C_M) > I(C_1, \cdots, C_{M+1})
\end{aligned}
\tag{4.2}
$$

The time complexity of this algorithm is $O(2^K M)$, where K is the size of the pre-built concept corpus, and M $(1 < M \leq K)$ is a variable to depict the number of concepts in a concept bundle. Generally, we can tune the value of threshold to reduce the computational time. A larger value for threshold will stop the algorithm earlier.

4.2.2 Learning Concept Bundle Classifier

4.2.2.1 Concept Utility Estimation

Intuitively, the M concepts in the concept bundle $(C_1, C_2, \cdots, C_M)$ may have different utilities for learning the concept bundle classifier. Take the concept bundle ("children", "sitting", "classroom") as an example. While most of the relevant samples in the primitive concept "classroom" may be relevant to the concept bundle, and it may not be true for concepts "children" and "sitting".

Thus, the classifier of the concept "classroom" should have a larger contribution to the concept bundle as compared to the other concepts. Here, we estimate the concept utilities in advance of learning the concept bundle.

For each concept C_m in the concept bundle $(C_1, C_2, \cdots, C_M)$, we collect the tags of the videos retrieved by C_m in Section 4.2.1 to form a tag document. As a result, we obtain M tag documents for the M concepts. We then regard $(C_1, C_2, \cdots, C_M)$ as a term and compute its term frequency and inverse document frequency (*tf-idf*) [MRS09] scores with respect to all the M documents. These M *tf-idf* scores are then normalized as $\{\beta_m\}_{m=1}^{M}$ $(0 \leq \beta_m \leq 1)$ which reflect the importance of the M primitive concepts with respect to the concept bundle. The larger value of β_m indicates the concept C_m is more important in the $(C_1, C_2, \cdots, C_M)$.

4.2.2.2 Classification Algorithm

Given a concept bundle $(C_1, C_2, ...C_M)$, we denote $\mathcal{D}_m = \{\mathbf{x}_i^m, y_i^m\}_{i=1}^{N_m}$, $\mathcal{D}_\mathcal{O} = \{\mathbf{x}_i^O, y_i^O\}_{i=1}^{N_O}$ as the training samples of the primitive concept C_m $(1 \leq m \leq M)$ and the concept bundle respectively. N_m, N_O are the numbers of samples in $\mathcal{D}_m$ and $\mathcal{D}_\mathcal{O}$ respectively; $\mathbf{x}_i^m, \mathbf{x}_i^O \in \mathbb{R}^d$ are the d-dimensional feature vectors of the i-th samples; and $y_i^m, y_i^O \in \{1, -1\}$ are binary labels indicating $\mathbf{x}_i^m$, $\mathbf{x}_i^O$ to be a relevant or irrelevant sample. Our goal is to learn a classifier $f(\mathbf{x})$ for the concept bundle $(C_1, C_2, ...C_M)$ based on the training samples $\{\mathcal{D}_m\}_{m=1}^{M} \bigcup \mathcal{D}_\mathcal{O}$.

Since the concept bundle $(C_1, C_2, ...C_M)$ is an intersection among its M primitive concepts in terms of both semantic and data exemplars, we assume that the concept classifiers $\{f_m(\mathbf{x})\}_{m=1}^{M}$ of the M primitive concepts usually share a common area in the decision space, which characterizes the regions of the concept bundle. We thus formulate each $f_m(\mathbf{x})$ as a combination of two parts: a private classifier and a communal classifier. The communal classifier is actually the concept bundle classifier $f(\mathbf{x})$, which is shared among all its constituent primitive concepts. Here, we formulate each classifier as:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{w}\phi(\mathbf{x}) \\ f_m(\mathbf{x}) &= \mathbf{w}_m\phi(\mathbf{x}) + \beta_m\mathbf{w}\phi(\mathbf{x}) \end{aligned} \tag{4.3}$$

where $\mathbf{w}$ and $\{\mathbf{w}_m\}_{m=1}^M$ are model parameters; $\phi(\cdot)$ is the feature map function projecting the original feature $\mathbf{x}$ into the transformed space; and β_m is the weight parameter described in Section 4.2.2.1 that indicates the utility of the m-th primitive concept in learning the concept bundle classifier $f(\mathbf{x})$.

Aforementioned, learning concept bundles suffers from the sparse relevant sample problem. Here, we consider two cases: (a) there is no relevant sample for a given concept bundle (the training data for the concept bundle is unavailable); and (b) the training data for the concept bundle is available. For the sake of simplicity, we only introduce the formulation for case (b), and the corresponding algorithm for case (a) is actually the unsupervised counterpart, which can be inferred by deleting the terms related to the samples in $\mathcal{D}_{\mathbb{O}}$. We derive $f(\mathbf{x})$ by jointly learning the concept classifiers $\{f_m(x)\}_{m=1}^M$ as:

$$\begin{aligned} \min_{\{\mathbf{w}_m\}_{m=1}^M, \mathbf{w}} \; & \tfrac{\lambda M}{2}||\mathbf{w}||^2 + \tfrac{1}{2}\sum_{m=1}^{M}||\mathbf{w}_m||^2 + C(\sum_{m=1}^{M}\sum_{i=1}^{N_m}\xi_i^m + \sum_{i=1}^{N_O}\xi_i) \\ s.t. \quad & y_i^m(\mathbf{w}_m\phi(\mathbf{x}_i^m) + \beta_m\mathbf{w}\phi(\mathbf{x}_i^m)) \geq 1 - \xi_i^m \\ & y_j^O\mathbf{w}\phi(\mathbf{x}_j^O) \geq 1 - \xi_j, \quad \xi_i^m, \xi_i \geq 0 \\ & m = 1, 2, ...M \quad i = 1, 2, ...N_m \quad j = 1, 2, ...N_O \end{aligned} \tag{4.4}$$

where the first and second terms are the two regularization terms to prevent "overfitting", and the third term is the penalty on the training errors. λ and C are the tradeoff parameters, and ξ_i^m is the slack variable. The objective function in Eq. (4.4) can be rewritten as the following (primal) Lagrangian function:

$$\begin{aligned} L_P = & \tfrac{\lambda M}{2}||\mathbf{w}||^2 + \tfrac{1}{2}\sum_{m=1}^{M}||\mathbf{w}_m||^2 + C(\sum_{m=1}^{M}\sum_{i=1}^{N_m}\xi_i^m + \sum_{i=1}^{N_O}\xi_i) \\ & - \sum_{m=1}^{M}\sum_{i=1}^{N_m}\alpha_i^m(y_i^m(\mathbf{w}_m\phi(\mathbf{x}_i^m) + \beta_m\mathbf{w}\phi(\mathbf{x}_i^m)) - (1 - \xi_i^m)) \\ & - \sum_{i=1}^{N_O}\alpha_i(y_i^O\mathbf{w}\phi(\mathbf{x}_i^O) - (1 - \xi_i)) - \sum_{i=1}^{N_O}\mu_i\xi_i - \sum_{m=1}^{M}\sum_{i=1}^{N_m}\mu_i^m\xi_i^m \end{aligned} \tag{4.5}$$

where $\mu_i^m, \mu_i > 0$, $\alpha_i^m, \alpha_i \geq 0$ are the Lagrange multipliers. We minimize the Lagrangian function by setting its derivative with respect to $\mathbf{w}$, $\mathbf{w}_m$, ξ_i^m and ξ_i

to zero respectively. This results in:

$$\begin{aligned}
\mathbf{w}_m &= \sum_{i=1}^{N_m} \alpha_i^m y_i^m \phi(\mathbf{x}_i^m) \\
\mathbf{w} &= \begin{cases} \frac{\sum_{m=1}^{M} \beta_m \mathbf{w}_m}{\lambda M} & for\ case\ (a) \\ \frac{\sum_{m=1}^{M} \beta_m \mathbf{w}_m + \sum_{i=1}^{N_O} \alpha_i y_i \phi(\mathbf{x}_i^O)}{\lambda M} & for\ case\ (b) \end{cases} \\
\alpha_i^m &= C - \mu_i^m, \qquad \alpha_i = C - \mu_i
\end{aligned} \tag{4.6}$$

In Eq. (4.6), $\mathbf{w}_m$ is a linear combination of the training samples $(\mathbf{x}_i^m, y_i^m)_{i=1}^{N_m}$ which is the same as the expression in the typical SVM approach. In the unsupervised learning algorithm for case (a), $\mathbf{w}$ is a linear combination of the training samples of the M primitive concepts. This expression of $\mathbf{w}$ is similar with the expression of the Combine Weighted fusion [AHO07]. However, the primitive concept classifiers here are jointly learned rather than independently learned as in [AHO07]. Additionally, the primitive concepts with large β_m will contribute significantly to the concept bundle classifier $\mathbf{w}$. In the supervise learning for case (b), the expression of $\mathbf{w}$ consists of the classifiers of the primitive concepts as well as the training samples from the concept bundle. Substituting Eq. (4.6) into its Lagrangian function, we get the Lagrange dual objective function expressed in a matrix form as:

$$L(\mathbf{A}) = \arg\max_{\mathbf{A}} [-\mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{1}\mathbf{A}] \tag{4.7}$$

where

$$\mathbf{A} = \begin{bmatrix} \alpha_1^1 & ... & \alpha_{N_1}^1 & ... & \alpha_1^M & ... & \alpha_{N_M}^M & \alpha_1 & ... & \alpha_N \end{bmatrix}^T$$

$$\mathbf{1} = \begin{bmatrix} 1 & 1 & 1 & ... & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} \frac{1}{2}(\frac{\beta_1^2}{\lambda M} + 1)\kappa_{11} & ... & \frac{1}{2\lambda M}\beta_1\beta_M\kappa_{1M} & \frac{1}{2\lambda M}\beta_1\kappa_{1O} \\ ... & ... & ... & ... \\ \frac{1}{2\lambda M}\beta_M\beta_1\kappa_{M1} & ... & \frac{1}{2}(\frac{\beta_M^2}{\lambda M} + 1)\kappa_{MM} & \frac{1}{2\lambda M}\beta_M\kappa_{MO} \\ \frac{1}{2\lambda M}\beta_1\kappa_{O1} & ... & \frac{1}{2\lambda M}\beta_M\kappa_{OM} & \frac{1}{2\lambda M}\kappa_{OO} \end{bmatrix}$$

$$\kappa_{ij} = \begin{bmatrix} y_1^i \mathcal{K}(\mathbf{x}_1^i, \mathbf{x}_1^j) y_1^j & \dots & y_1^i \mathcal{K}(\mathbf{x}_1^i, \mathbf{x}_{N_j}^j) y_{N_j}^j \\ y_2^i \mathcal{K}(\mathbf{x}_2^i, \mathbf{x}_1^j) y_1^j & \dots & y_2^i \mathcal{K}(\mathbf{x}_2^i, \mathbf{x}_{N_j}^j) y_{N_j}^j \\ \dots & \dots & \dots \\ y_{N_i}^i \mathcal{K}(\mathbf{x}_{N_i}^i, \mathbf{x}_1^j) y_1^j & \dots & y_{N_i}^i \mathcal{K}(\mathbf{x}_{N_i}^i, \mathbf{x}_{N_j}^j) y_{N_j}^j \end{bmatrix}$$

where $\mathcal{K}(,)$ is a kernel function and κ_{ij} represents the kernel matrix of the training samples from the concept i and j. The optimization problem expressed in Eq. (4.7) can be solved using SMO algorithm [BLJ04]. The time complexity to solve Eq. (4.7) is $O(N^3)$, where N is the total number of all the training samples from primitive concepts and concept bundles.

4.3 Experimental Results

We conducted experiments on two video datasets. The first one is "TV08" dataset (see section 3.1.1), and the other one is "YT10" dataset (see section 3.2.1). On these two datasets, we used the training sets to train the classifiers of the concept bundles, and used the testing sets to test the performance of the learned concept bundle classifiers.

Based on the primitive concepts, we selected informative concept bundles according to their informativenesses measured in Eq. (7.5). We conducted this selection on the 374 primitive concepts of "TV08" dataset and 41 primitive concepts of "YT10" dataset. In calculating the informativeness of a concept bundle, we set the weight α in Eq. (7.5) to 0.5 to equally emphasize the co-occurrence of primitive concepts and users' interest on this concept bundle. For informative concept bundle selection, we set the threshold as the average informativeness of all the bundle candidates. In particular, the threshold is set to 0.25 for "TV08" dataset and 0.2 for "YT10" dataset. As a result, we obtained 40 concept bundles on "TV08" dataset (see Table 4.1), and 38 concepts bundles on "YT10" dataset (see Table 4.2).

From Tables 4.1 and 4.2, it is evident that some concept bundles are quite useful to model complex query, such as ("street", "nighttime") for query "a street scene at night" and ("computer", "office") for query "one or more people at a table or desk with a computer visible" on "TV08" dataset. However, some concept

Table 4.1: The 40 informative concept bundles on "TV08" dataset (The concept bundles in bold are evaluated in our experiment)

1	classroom,room	21	meeting,room
2	glasses,sunglasses	22	building,cityscape
3	basketball,stadium	23	**office,person**
4	**airplane,sky**	24	airplane,military
5	hand,handshaking	25	dance,street
6	bathroom,room	26	bird,sky
7	**building,sky**	27	house,room
8	athlete,sports	28	bus,driver
9	police,protester	29	car,police
10	**bridge,waterway**	30	**nighttime,street**
11	**street,vehicle**	31	face,hand
12	**crowd,outdoor**	32	mountain,snow
13	outdoor,sport	33	**animal,person**
14	swimmer,swimming	34	**computer,office**
15	animal,dog	35	landscape,sky
16	**flower,vegetation**	36	**dining_room,food**
17	clouds,sky	37	**computer_or_television _screens,person**
18	car,vehicle	38	desert,landscape
19	airplane,airport	39	mountain,sky
20	**ship,waterway**	40	car,desert

bundles are not very useful. For example, ("animal", "dog"), ("car", "vehicle") etc, are actually equivalent to primitive concepts since they are synonym or hyponym conveying similar semantics. Such concept bundles can be easily identified by Wordnet [Fel98]. Thus, we employed Wordnet to filter out such concept bundles. After that, we selected a subset of concept bundles in Tables 4.1 and 4.2 to evaluate since the labeling of ground truth is labor-intensive. These selected concept bundles are those that will be used in our search task. For example, in Table 4.1, the third concept bundle ("basketball", "stadium") is not selected since it is not related to any "TV08"'s queries. As a result, we selected 13 concept bundles (in bold font) in Table 4.1 and 22 concept bundles (in bold font) in Table 4.2.

On "TV08" dataset, we directly used Columbia374 primitive concept classifiers, while on "YT10" dataset we trained the primitive concept classifiers using the Support Vector Machine (SVM). In the SVM algorithm, we used the empirically successful Gaussian kernel $exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$ as the kernel function where γ is the scaling parameter. A fivefold cross-validation process was conducted on the training set to determine the parameters.

As described in Section 4.2.2.2, we have considered two cases in our classification algorithm: Unsupervised Learning (UL) without using the training samples for the concept bundles and Supervised Learning (SL) by using the training samples. Although all the concept bundles have training samples in our datasets, we still trained two kinds of the classifiers by the UL and SL approaches for each selected concept bundle in order to make a complete comparison with the following four existing methods:

- Support Vector Machine (SVM): This approach directly builds an SVM classifier for each concept bundle based on its training samples without using the primitive concepts. The Gaussian kernel is adopted and the parameters are estimated through a fivefold cross-validation process.

- Concept Fusion (CF) [AHO07]: This approach generates the detection result of a concept bundle by fusing the individual results from its primitive concept classifiers. Here we adopt the Combine Weighted fusion operation, and the fusion weight of each primitive concept is equal to its utility value

Table 4.2: The 38 informative concept bundles on "YT10" dataset (The concept bundles in bold are evaluated in our experiment)

1	**lady,telephone**	20	horse,race
2	**baby,laughing**	21	motorcycle,ride
3	car,race	22	**person,television**
4	**fighting,street**	23	**lady,office**
5	**dance,wedding**	24	baby,dance
6	race,river	25	cat,fighting
7	crash,motorcycle	26	car,race,river
8	**police,protester**	27	**fire,vehicle**
9	dance,street	28	**children,singing**
10	**horse,riding**	29	**interview,lady**
11	**motorcycle,race**	30	crash,vehicle
12	race,street	31	motorcycle,street
13	**car,race,crash**	32	**astronaut,moon land**
14	kitchen,sea	33	interview,protest
15	**lion,hunting,zebra**	34	**cooking,kitchen**
16	**cat,mouse**	35	**fighting,soccer**
17	dance,lady	36	**basketball,street**
18	**house,tree**	37	**beach,sea**
19	car,police	38	**flood,river**

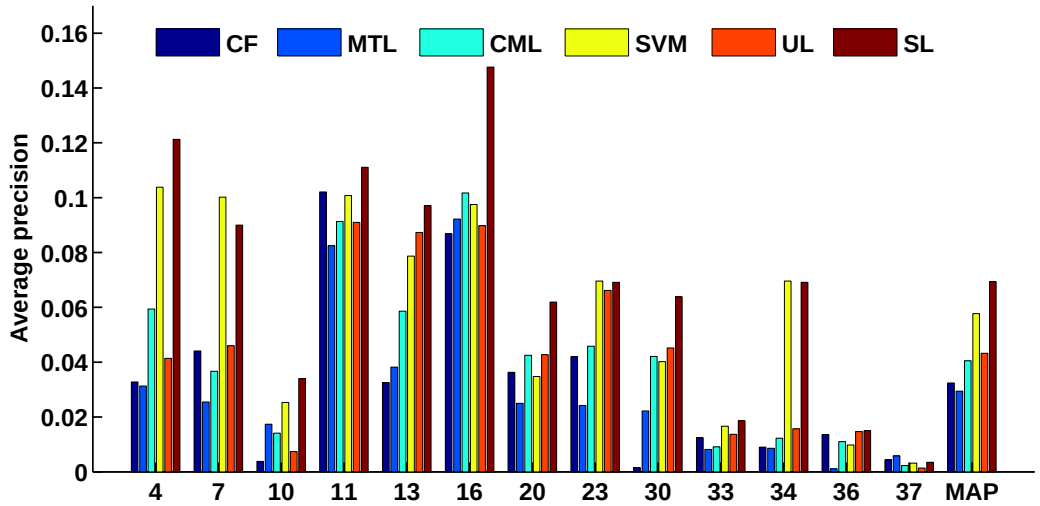


Figure 4.1: The performance on 13 concept bundles of "TV08" dataset as measured by AP@1000

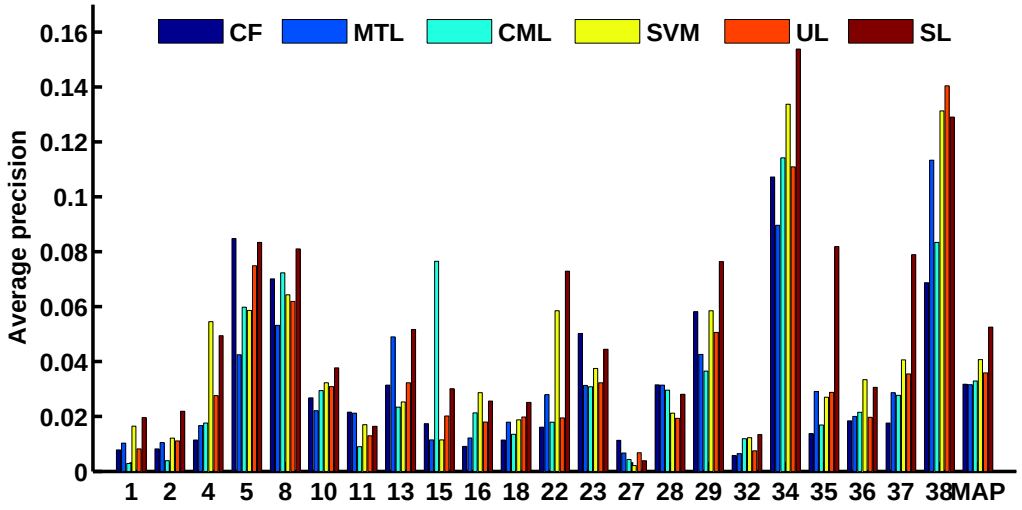


Figure 4.2: The performance on 22 concept bundles of "YT10" dataset as measured by AP@1000

with respect to the concept bundle.

- Multi-label Learning Approach (CML) [QHR+07]: For each concept bundle, this approach jointly learns its involved primitive concepts on their training data, and outputs the predicted probabilities on all the primitive concepts for each testing sample. We then generates the detection result of the concept bundle using the CF approach.

- Multi-Task Learning Approach (MTL) [AZ05]: This approach takes each involved primitive concept of a concept bundle as a learning task, and jointly learns these tasks with a shared parameter part. In the implementation, we adopt the modified Huber loss function as suggested in [AZ05], and solve the optimization problem by the stochastic gradient descent method. After obtaining the results of the involved primitive concepts of a concept bundle, it generates the detection result of the concept bundle using the CF approach.

The performance is measured by the widely used non-interpolated Average Precision (AP) [TRE08]. We averaged the APs over all the concept bundles to obtain the Mean Average Precision (MAP) as a measure of the overall performance.

Figures 4.1 and 4.2 illustrate the performance of our UL and SL methods as compared to that of CF, SVM, CML and MTL approaches. The following observations are obtained:

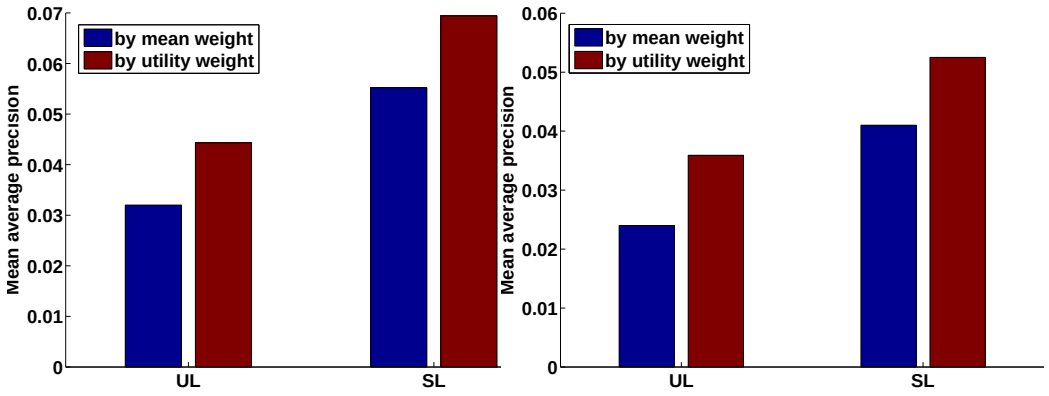
- The best performance is achieved by the SL approach with a MAP of 0.069 on "TV08" dataset and 0.053 on "YT10" dataset. As compared to the results of many concept detection techniques which could achieve the MAP of above 0.1 for the primitive concepts, the performance of concept bundle learning is much lower. This is because the relevant samples of concept bundle are much fewer than that of the primitive concepts.

- The SVM approach achieves a MAP of 0.058 on "TV08" dataset and 0.041 on "YT10" dataset. Compared to the SVM approach, the improvement by the SL approach is about 19.0% and 29.3% on "TV08" and "YT10" datasets respectively. This improvement re-affirms our view that jointly learning the concept bundles and their primitive concepts can overcome the sparse relevant sample problem, and thus be able to achieve better performance.

- The UL approach only has a MAP of 0.043 on "TV08" and 0.036 on "YT10" dataset, with a corresponding 25.0% and 11.8% performance degradation as compared to the SVM approach. This is because our UL approach is unsupervised and does not require the training samples of concept bundles.

- The other approaches CF, MTL, CML all have a poorer performance as compared to UL, SVM and SL approaches. This is because CF, MTL, CML approaches focus on learning the individual primitive concepts instead of concept bundle, and the results of the concept bundle are actually generated by the fusion operation. As discussed before, simply aggregating primitive concepts cannot interpret the concept bundle well.

To evaluate the effectiveness of concept utility in our UL and SL approach, Figure 4.3 shows the performance comparison of the UL and SL approaches by using our utility weights or simply adopting mean weights ($\beta = \frac{1}{m}$ in Eq.(4.3)). We can see that using utility weights leads to better performance. This demonstrates that the concept utility could correctly estimate the degree of relatedness



(a) "TV08" Dataset (b) "YT10" Dataset

Figure 4.3: The effectiveness of concept utility in UL and SL

of primitive concepts with respect to the concept bundle. Taking the concept bundle ("person", "office") on "TV08" dataset as example, the utilities for the primitive concepts "person" and "office" are 0.13 and 0.87 respectively. This enables the concept bundle ("person", "office") to achieve an AP of 0.069 by the SL approach, instead of an AP of 0.035 by the SL approach using the mean weights.

4.4 Conclusion

In this chapter, we proposed to learn a higher-level semantic descriptor named "concept bundle" to facilitate video search for complex queries. A concept bundle is defined as a composite semantic concept that integrates multiple primitive concepts. To ensure the informativeness of concept bundles, we devised a selection process based on concept bundle frequency in the suggested queries by Web video search engine and the concept co-occurrence in the tags of Web videos. We then proposed a multi-task SVM algorithm to build the concept bundle classifier by jointly learning its involved primitive concept classifiers. The experiments were conducted on "TV08" and "YT10" datasets and demonstrated that our multi-task SVM algorithm achieved promising performance as compared to the state-of-the-art approaches in modeling concept bundles.

The proposed concept bundle provides a solid support for learning complex query in semantic video search. In the next chapter, we will introduce an au-

tomatic semantic video search approach for complex queries by using concept bundles.

Chapter 5

Bundle-based Automatic Semantic Video Search

5.1 Introduction

In Chapter 4, we propose a higher-level semantic descriptor named "concept bundle". Carrying higher-level semantics, concept bundle is expected to deliver more precise descriptions of video contents, and thus better meet the video search requirement in a finer granularity. For example, in Figure 5.1, the complex query "persons dancing in the wedding" can be better answered by using the concept bundle ("dance", "wedding") than the typical semantic video search approach which uses only the primitive concepts.

However, to achieve a good search performance for complex query, we need to select a proper set of concept bundles to interpret the users' query in addition to primitive concepts. In this chapter, we propose an concept selection strategy to map the query to related primitive concepts and concept bundles by considering their classifier performance and semantic relatedness with respect to the query. The final search results are generated by fusing the individual results from these selected primitive concepts and concept bundles.

We evaluate the proposed approaches on two video datasets: TRECVID 2008 and YouTube 2010 datasets. Compared to the state-of-the-art methods, the use of concept bundles can characterize the complex queries well and improve the

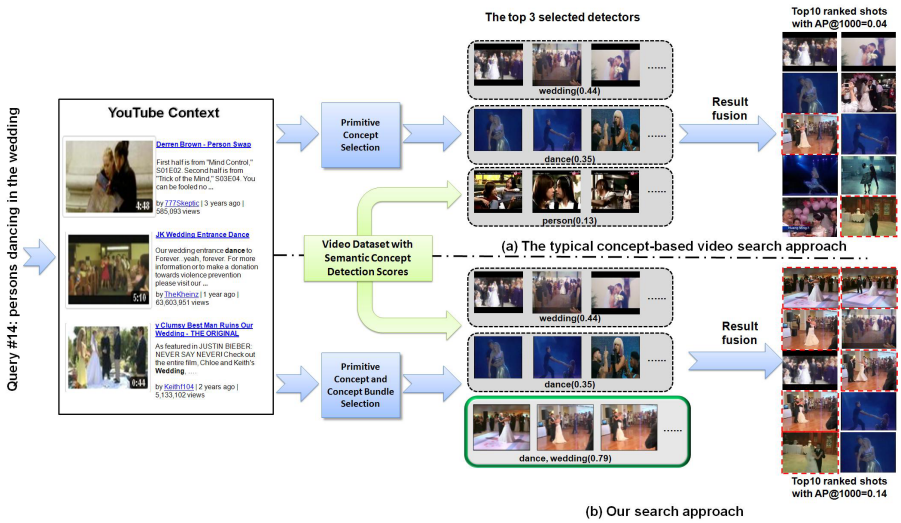


Figure 5.1: Illustration of the search procedure in the traditional semantic video search (part (a)) and our search approach (part (b)) for the complex query "persons dancing in the wedding". In our search approach, the selected concept bundle ("dance", "wedding") is semantically closer to the query. We list the top 10 retrieved video shots by these two approaches, where the rank lists are ordered from left to right and top to bottom (positive samples are marked in red boxes).

search performance by at least 37.5% and 52% on TRECVID 2008 and YouTube 2010 datasets respectively.

5.2 Bundle-based Video Search

Given a query Q, the typical semantic video search approach first maps this query into related primitive concepts before computing the relevance of each video entry with respect to the query. In our search model, we map the query into related primitive concepts and concept bundles. For the sake of simplicity, we use the word "bundle" to imply both a primitive concept and a combined semantic concept (concept bundle) in this section.

5.2.1 Mapping Query to Bundles

5.2.1.1 Formulation

Based on the pre-built bundle corpus $\mathcal{S} = \{\mathcal{C}_1, ..., \mathcal{C}_K\}$, we need to select related bundles $\{\mathcal{C}_1, ..., \mathcal{C}_L\}$ $(L < K)$ for each query Q. Our selection approach selects related bundles (denoted as a set $sub(\mathcal{S})$) from $\mathcal{S}$ according to two criterions: (1) $sub(\mathcal{S})$ should preferably contain the bundles with high semantic relatedness with respect to query Q; and (2) $sub(\mathcal{S})$ should introduce as little errors as possible. We make a tradeoff between these two criterions by a weight parameter C, and the formulation is expressed as:

$$\arg\min_{sub(\mathcal{S})} C * (1 - Sem(sub(\mathcal{S}))) + (1 - C) * Er(sub(\mathcal{S})) \tag{5.1}$$

where $Sem(sub(\mathcal{S}))$ measures the semantic relatedness of the bundles in $sub(\mathcal{S})$ with respect to the query Q, and $Er(sub(\mathcal{S}))$ is the errors produced by the classifiers of the bundles in $sub(\mathcal{S})$.

5.2.1.2 Semantic Relatedness Estimation

Different concepts have different semantic relatedness to query Q, and this semantic relatedness can be estimated by the text matching score [CHJ+06]. In our approach, we use the *tf-idf* score between the query and concept to represent this text matching score. For each query Q, we first generate a parse tree [MS99] by using OpenNLP [1]. We then select the nouns and verbs from the parse tree as the salient words since they are found to be more important than articles, adjectives or adverbs [WMC09]. Finally we take these selected salient words as a term and compute its *tf-idf* scores with respect to the K video tag documents of the bundles in $\mathcal{S}$, where each tag document is a collection of the video tag files downloaded from YouTube website. We normalize all the *tf-idf* scores in $\mathcal{S}$ as the bundle utilities $\{\beta_i\}_{i=1}^{|\mathcal{S}|}$. The semantic relatedness of $sub(\mathcal{S})$ is defined as the sum of the bundle utilities in $sub(\mathcal{S})$:

$$Sem(sub(\mathcal{S})) = \sum_{s_i \in sub(\mathcal{S})} \beta_i \tag{5.2}$$

[1] http://incubator.apache.org/opennlp/

5.2.1.3 Error Estimation

The performances of the pre-built bundle classifiers are quite different, and we want to select relevant classifiers that introduce as little errors as possible. Let $Er(sub(\mathcal{S}))$ be the misclassification probability of a sample $\mathbf{x}$ by the classifiers in $sub(\mathcal{S})$, which we express as:

$$Er(sub(\mathcal{S})) = \sum_{y} P(\hat{y} \neq y|\mathbf{x})P(y) \tag{5.3}$$

where $P(y)$ is the prior probability of sample $\mathbf{x}$ with label y, and $P(\hat{y} \neq y|\mathbf{x})$ is the misclassification probability of sample $\mathbf{x}$ ($\hat{y}$ is the predicted label). In our problem, we define two labels: $y \in \{1, -1\}$, where $y = 1$ means that $\mathbf{x}$ is a positive sample for query Q, and $y = -1$ otherwise. Based on these two labels, we transform Eq. (5.3) into Eq. (5.4) as:

$$\begin{aligned} Er(sub(\mathcal{S})) = {} & P(\hat{y} = -1|\mathbf{x}, y = 1)P(y = 1) \\ & + P(\hat{y} = 1|\mathbf{x}, y = -1)P(y = -1) \end{aligned} \tag{5.4}$$

Let $\mathbf{y^b} \in \mathcal{Y} = \{1, -1\}^L$ denote the L dimensional label vector of a sample $\mathbf{x}$ with respect to the bundles of $sub(\mathcal{S})$, where L is the number of bundles in $sub(\mathcal{S})$. A sample is positive if it satisfies all the bundles in $sub(\mathcal{S})$, or negative even if it is just irrelevant to one bundle in $sub(\mathcal{S})$. Thus, the label of a positive sample is $\mathbf{y^b} = \{1\}^L$, and that of a negative sample is $\mathbf{y^b} \in \mathcal{Y}/\{1\}^L$. As a result, we express the probabilities $P(\hat{y} = -1|\mathbf{x}, y = 1)$ and $P(\hat{y} = 1|\mathbf{x}, y = -1)$ as:

$$\begin{aligned} P(\hat{y} = -1|\mathbf{x}, y = 1) &= 1 - \prod_{i}^{L} P(\hat{y}_i^b = 1|\mathbf{x}, y_i^b = 1) \\ P(\hat{y} = 1|\mathbf{x}, y = -1) &= \max_{\mathbf{y^b} \in \mathcal{Y}/\{1\}^L} \prod_{i}^{L} P(\hat{y}_i^b = 1|\mathbf{x}, y_i^b) \end{aligned} \tag{5.5}$$

where y_i^b is the i-th label of $\mathbf{y^b}$, and $\hat{y}_i^b$ is the predicted label by the i-th bundle classifier in $sub(\mathcal{S})$. In Eq.(5.5), for a positive sample, its every individual label y_i^b is 1, thus $P(\hat{y} = -1|\mathbf{x}, y = 1)$ can be directly calculated; For a negative sample, since its individual label y_i^b is uncertain, we set $P(\hat{y} = 1|\mathbf{x}, y = -1)$ to be the maximum misclassification probability among all the possible label

assignments($\mathbf{y^b} \in \mathcal{Y}/\{1\}^L$).

To estimate the unknown probabilities in Eqs. (5.4) and (5.5), we resort to a validation dataset. We estimate the conditional probabilities $P(\hat{y}_i^b = 1|\mathbf{x}, y_i^b = 1)$ and $P(\hat{y}_i^b = 1|\mathbf{x}, y_i^b = -1)$ to be the proportion of positive (negative) samples of the i-th bundle being classified as positive on the validation set. While for the prior probabilities $P(y = 1)$ or $P(y = -1)$, we estimate them to be the proportion of positive(negative) samples over the total number of all samples on the validation set.

5.2.1.4 Implementation

In our approach, we employ an optimization function to select a set of related bundles to interpret a query. However, the computation cost will be high if we try all the possible sets (the time complexity is $O(2^K)$). Here, we develop a greedy algorithm to speed up the selection procedure. We first select a bundle from $\mathcal{S}$ with the minimal value of Eq. (5.1), and then incrementally add a bundle that maximally reduces the value of Eq. (5.1). This incremental process is stopped when the value of Eq. (5.1) cannot be further reduced by the remaining bundles. In our implementation, the sub-bundles of an selected bundle will no longer be evaluated and selected by the algorithm. The rationale is that a bundle is semantically closer to the query than its sub-bundles. By using the greedy algorithm, the time complexity is reduced to $O(K|sub(\mathcal{S})|))$. We summarize the process of mapping-query-to-bundles in Algorithm 1.

5.2.2 Fusion

Different from the typical video search approach that fuses the results from related primitive concepts, our approach generates the final results by fusing the individual results from the related bundles. Given a selected bundle set $sub(\mathcal{S})$, with each bundle has a classifier f^l, we compute the relevance score of each keyframe with respect to the complex query as

$$Score(x) = \sum_{l=1}^{|sub(\mathcal{S})|} \beta_l \frac{1}{1 + e^{-f^l(\mathbf{x})}} \tag{5.6}$$

Algorithm 1 Mapping-query-to-bundles

Input: The query Q, the pre-built bundle corpus $\mathcal{S} = \{\mathcal{C}_k\}_{k=1}^{K}$, the weight parameter C
Output: the related bundle set $sub(\mathcal{S})$
Process:
$\{\beta_k\}_{k=1}^{K} \longleftarrow$Semantic_Relatedness_Estimation(Q, $\mathcal{S}$)
$sub(\mathcal{S}) = \varnothing$
while there exists an unselected bundle s_i in $\mathcal{S}$ such that the value of Eq. (5.1) can be reduced **do**
    $\arg\min_{s_i} C * (1 - Sem(sub(\mathcal{S}) \cup s_i)) + (1 - C) * Er(sub(\mathcal{S}) \cup s_i)$
    $sub(\mathcal{S}) = sub(\mathcal{S}) \cup s_i$
end while
return $sub(\mathcal{S})$.

where β_l is the utility value of the l-th bundle. The calculation of β_l is described in Section 5.2.1.2. Finally, we use the relevance scores of all the keyframes to rank the final search results.

5.3 Experimental Results

The experiments were conducted two video datasets: the "TV08" dataset (see section 3.1.1), and the "YT10" dataset (see section 3.2.1).

By using the concept bundle learning approach in Chapter 4, we have built 13 concept bundles on "TV08" dataset and 22 concept bundles on "YT10" dataset (see the bold font in Table 4.1, 4.2). Based on these concept bundles as well as the primitive concepts (the concept Columbia374 on "TV08" dataset and 41 primitive concepts on "YT10" dataset), we evaluate the search performance of our bundle-based video search approach on the testing sets of these two datasets.

We conducted experiments on all the 48 queries on "TV08" dataset and 20 queries on "YT10" dataset. In our experiments, we employed the SL algorithm in chapter 4 to train the classifiers for concept bundles. The weight parameter C in Eq.(5.1) was set to 0.8 due to its best performance as shown in Table 5.2. The probabilities in section 5.2.1.3 were obtained through evaluating the classification results of the classifiers on the validation datasets.

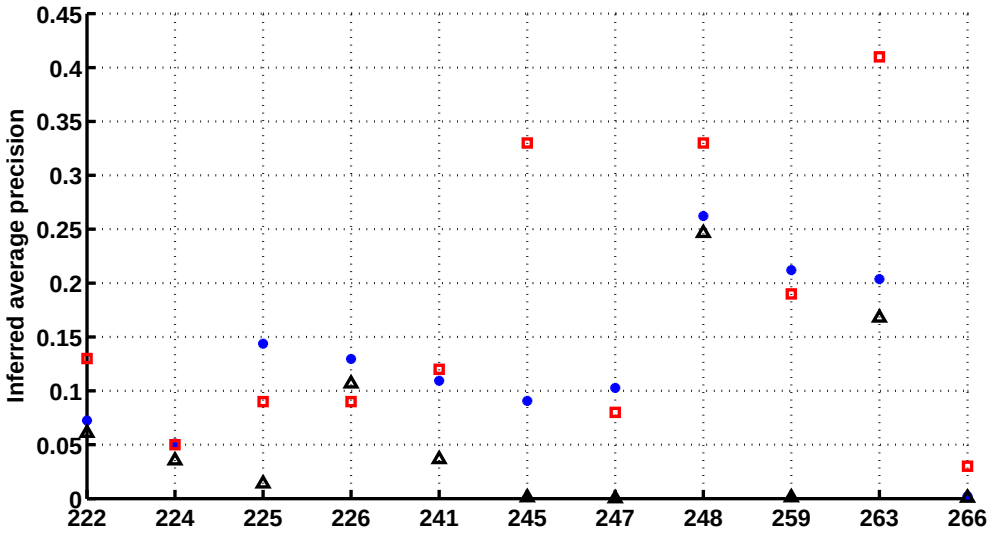


Figure 5.2: The detailed performance of the selected 11 queries on "TV08" dataset as measured by inferred AP@1000, where the rectangle is the best performance achieved by the official submissions on "TV08" search task, star and triangle are the performance achieved by our search approach using and not using concept bundles respectively

In the first experiment, we evaluated the effectiveness of concept bundles in the overall search performance. We compared the search performance on two cases by: (1) mapping the query to related primitive concepts and concept bundles (using the concept bundles); and (2) mapping the query to only related primitive concepts (without using the concept bundles). Here, the search without concept bundles is performed using the same approach as in Section 5.2, but the concept bundles are not involved. Table 5.1 shows the comparison of search performance. On "TV08" dataset, we adopted the Inferred Average Precision (Inferred AP) as provided by TRECVID [TRE08] to evaluate the performance of each query and the overall performance is measured by the Inferred Mean Average Precision (Inferred MAP). The overall results demonstrate that, by using concept bundles, our search algorithm can improve the Inferred MAP on "TV08" dataset from 0.04 to 0.055 and the MAP on "YT10" dataset from 0.025 to 0.038. The relative improvements are 37.5% and 54.7%, respectively. These improvements are substantial in the context of video search, which is an extremely difficult task due to the high diversity of video content [SW09]. Here, statistical significance test

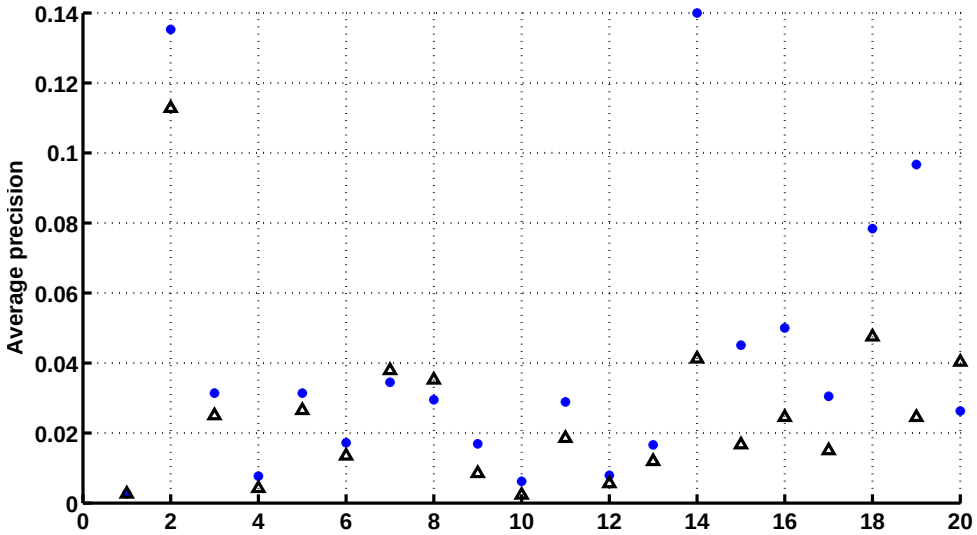


Figure 5.3: The detailed performance of the 20 queries on "YT10" dataset as measured by AP@1000, where the star and triangle are the performance achieved by our search approach using and not using concept bundles respectively

Table 5.1: The comparison of video search performance by using or not using concept bundles as measured by inferred MAP@1000 ("TV08") or MAP@1000 ("YT10")

	using concept bundles	not using concept bundles
"TV08"	0.055	0.040
"YT10"	0.038	0.025

is not performed to examine the results. The reason is that significance test is normally ineffective in evaluating video search systems due to the high variance among the query topics [HL05].

Figures 5.2 and 5.3 show the detailed performance of our search algorithm on each query by using and not using the concept bundles. Actually, not all the queries could be mapped to concept bundles. On "TV08" dataset, only 11 queries could be mapped to concept bundles in our experiments, while the remaining queries are modeled using only primitive concepts and could only achieve the same performance as the approach without using the concept bundles. Here, we only list the comparison results on these 11 queries. In Figure 5.2, on "TV08" dataset, our search approach using concept bundles outperforms that without us-

Table 5.2: The video search performance by using different weights C in Eq. (5.1)

	C=0	C=0.2	C=0.4	C=0.6	C=0.8	C=1
"TV08"	0.031	0.031	0.031	0.038	**0.055**	0.052
"YT10"	0.026	0.026	0.026	0.030	**0.038**	0.035

ing concept bundles on all the 11 queries. Among these queries, the performances of 5 queries (224, 225, 226, 247, 259) have substantially better performance as compared to the TRECVID official results. On "YT10" dataset, our search approach using the concept bundles performs better on 17 queries, and worse on 3 queries as compared to that without using the concept bundles. After analyzing these three queries (7, 8, 20), we find the search performance degradation is caused by the poor performance of the concept bundle classifiers (28, 38, 27 in Figure 4.2) used for these three queries.

Furthermore, we evaluated the performance variation with different weights C (see Eq. (5.1)). Table 5.2 shows the search results under different weights C. The best performance is achieved when $C = 0.8$. The smaller value of C will lead to a rapid performance degradation, and the larger value of C will slightly reduce the search performance. This result demonstrates that the semantic relatedness is more important than the classifier performance in concept selection. When $C < 0.4$, the concept selection results are dominated by selecting the high performance concepts, and thus the search performance is stable.

Finally, Table 5.3 lists the performance comparison between our search approach and the two state-of-the-art approaches [WN08; JNC09] on "TV08" dataset. In [WN08], a multi-level fusion framework is developed by considering the semantics, observability, reliability and diversity for classifier selection, while in [JNC09] the Flickr context is analyzed and used for classifier selection. In our approach, we utilize concept bundles to help learn the complex queries, and as a result we could achieve a substantial improvement. Figure 5.4 further compares our results with the official submissions on "TV08" dataset. Among all the 82 submissions, our approach, using only concept based query process, ranks third, while the top two TRECVID official submissions adopted a combination of concept and image/video examples matching. For example, the best performing system [TRE08] contains three modalities: text matching, semantic search, and

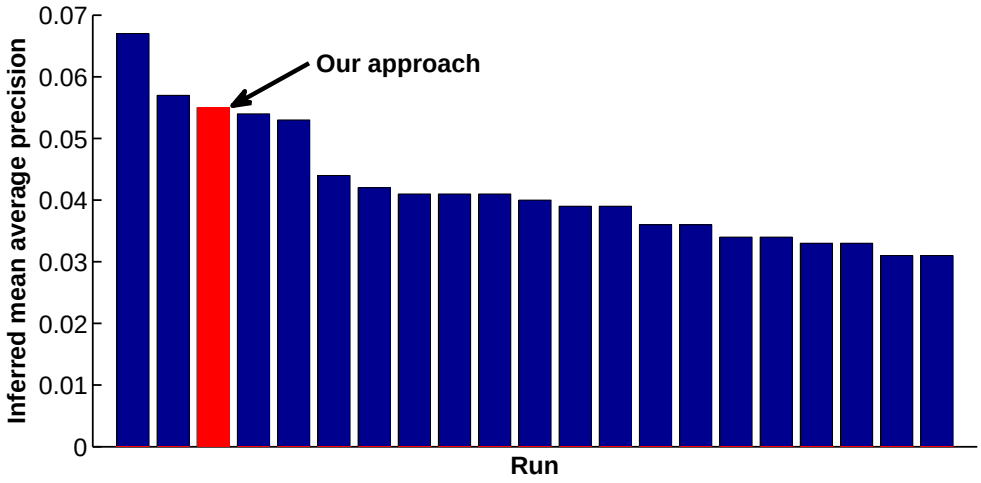


Figure 5.4: Inferred MAP comparison with the top-20 (out of 82) official submissions of the automatic video search task in TRECVID 2008

Table 5.3: The search performance comparison between our search approach and the state-of-the-art approaches on "TV08" dataset

	Wei et al. [WN08]	Jiang et al. [JNC09]	Our approach
TV08	0.042	0.050	0.055

image/video example matching.

5.4 Conclusion

In this chapter, we proposed an optimization function to map complex queries to concept bundles and primitive concepts. The mapping algorithm considers the semantic relatedness and the classifier performance. To improve the efficiency, we employed a greedy algorithm to approximately implement our approach. The experiments were conducted on "TV08" and "YT10" datasets. The results demonstrate that the concept bundles could characterize complex queries well and achieve promising search performance as compared to the state-of-the-art approaches.

The performance of the automatic semantic video search approaches is still

unsatisfactory. In the next chapter, we will introduce our interactive semantic video search approach to further enhance the search performance for complex queries.

Chapter 6

Related Sample based Interactive Semantic Video Search

6.1 Introduction

In chapter 5, the performance of the automatic semantic video search is still unsatisfactory especially for complex query. To improve search performance, we employ interactive semantic video search, which incorporates user in the search loop and has shown promising performance recently [SW09]. In interactive search process, the user is asked to label the retrieval list returned by the system. Based on user's annotation of relevant and irrelevant video segments, the system then performs relevance feedback to refine the search model for better retrieval results. By performing a few iterations, the retrieval is expected to return more and more relevant video segments. Generally, to ensure a quality interactive search, a reasonable amount of relevant samples are required to be annotated in the first few iterations. However, this may not be possible for complex queries, in which the relevant samples are usually rare or not ranked on the top of the result list. This insufficiency of relevant samples renders most relevance feedback techniques ineffective in interactive semantic video search.

To enhance the interactive search for complex queries, we propose to utilize a third class of video samples, i.e. "related samples", parallelling with relevant and irrelevant samples. Related samples refer to those video segments that are

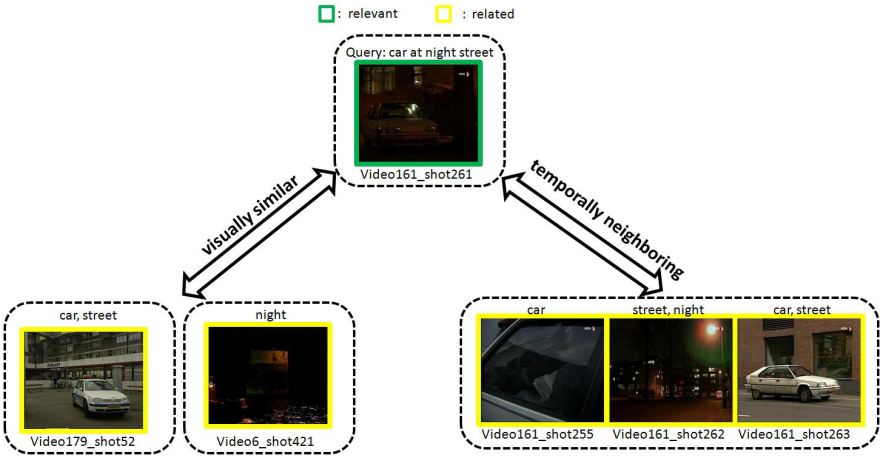


Figure 6.1: Exemplar related samples for the query "car at night street"

partially relevant to the query but do not satisfy the entire search criterion. As illustrated in Figure 6.1, the related samples of the query "car at night street" are the samples that contain the individual concepts "car", "night", or "street" rather than the scene of "car at night street". Compared to relevant samples which may be rare, related samples are usually more plentiful and easier to find in the search results. The advantages of exploring related samples are two-fold: First, the related and relevant video segments usually share similar visual content in part due to their semantic connection, so that the related samples are beneficial to the modeling of relevant samples. Second, since video content is temporally dynamic and continuous, the occurrence of related video segments is an indicator of the presence of relevant ones in the neighboring clips. Based on these motivations, in this chapter, we develop a visual-based ranking model that simultaneously exploits the visual information of relevant, related, and irrelevant samples and a temporal-based ranking model to utilize the temporal relationship between related and relevant samples. The search results are generated by fusing the results from these two models. Moreover, we develop an adaptive fusion method that optimizes the fusion weight based on user's labeling in each iteration of relevance feedback. The resultant optimal fusion can further boost the search performance.

We evaluate the proposed approach on two real-world video collections: TRECVID 2008 dataset, and YouTube 2011 dataset. The experimental results on both

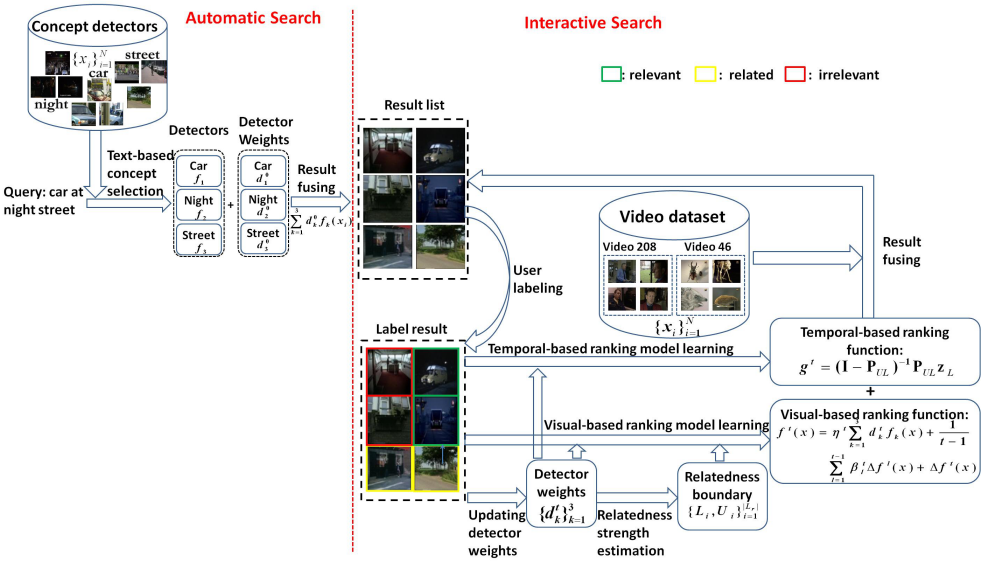


Figure 6.2: The framework of interactive semantic video search.

datasets demonstrate that our approach can achieve competitive search performance as compared to the state-of-the-art methods.

The main contributions of our approach can be summarized as follows:

- We propose to explore "related samples" to enhance interactive semantic video search.
- We develop a visual-based and a temporal-based ranking model to exploit the related samples, in parallel with the relevant and irrelevant samples.
- We develop an adaptive fusion strategy to optimally explore the two proposed ranking models.

6.2 Framework

Given a query Q, our target is to retrieve as many relevant video shots as possible, via a few feedback iterations. As aforementioned, a video shot is often represented by its keyframe [GKS00]. Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ denote a set of N keyframes, where $\mathbf{x}_i \in \mathbb{R}^d$ is the d-dimensional feature vector of keyframe i. In each iteration t

of interactive search, the search system presents users with the top N_t keyframes $\mathcal{L}^t = \{\mathbf{x}_i\}_{i=1}^{N_t}$ to label, and each keyframe is labeled as relevant, irrelevant or related. We use $\mathcal{Y}^t = \{y_i\}_{i=1}^{N_t}$ to denote the labels of these keyframes, where $y_i = 1$ indicates $\mathbf{x}_i$ is a relevant sample, and $y_i = -1$ means $\mathbf{x}_i$ is an irrelevant sample. For a related sample, we set y_i as its value of relatedness strength which refers to the relatedness degree of $\mathbf{x}_i$ with respect to the query. The estimation of relatedness strength will be introduced in next section.

Given a concept set $\mathcal{C}$, we pre-build concept classifiers $\{f_k\}_{k=1}^{|\mathcal{C}|}$. For each concept C_k in $\mathcal{C}$, we issue the concept name to Flickr website as a query and construct a tag document $\mathcal{T}_k$ by collecting the tags from the top 100 returned images. These tag documents $\{\mathcal{T}_k\}_{k=1}^{|\mathcal{C}|}$ are then indexed by Lucene [Luc], a widely used text search approach.

When a query is input, as Figure 6.2 shows, our interactive search system works as follows:

1. The system builds the concept bundle classifiers according to the approach discussed in chapter 4, and performs the automatic semantic video search based on the approach described in chapter 5.

2. At iteration t of interactive search, users label the top N_t samples as relevant, related or irrelevant samples. This gives rise to a labeled sample set $\{\mathcal{L}^t, \mathcal{Y}^t\}$.

3. The system updates concept weights $\{d_k^t\}_{k=1}^{K}$, estimates relatedness strength of related samples and learns a visual-based ranking model f^t.

4. The system learns a temporal-based ranking model g^t.

5. The system generates search results by fusing the individual results from the visual-based ranking model and the temporal-based ranking model.

6. Repeat from step 2) until the user is satisfied with the search results.

6.3 Approach

In this section, we first introduce related samples, and then elaborate the visual-based and temporal-based ranking models, as well as the adaptive fusion method.

6.3.1 Related Sample

The "related samples" refer to those video shots that are relevant to part of the query rather than the entire query. Here, we allow users to flexibly decide the condition of "part of the query". For example, in the query "a street scene at night", the related samples are those satisfying the concept "street" or "night". Another example is the query "one or more ships or boats in the water", the related samples are those containing "boat", "ship" or "water".

The advantage of using related samples is in two-fold. First, the related samples are usually visually similar with the relevant ones. This is so because the related samples tend to share visual contents with the relevant ones in part. Figure 6.3 (a) illustrates the visual similarities between the relevant and related samples of the query "one or more people with one or more horses". Second, videos carry temporally dynamic and continuous contents. The occurrence of a related sample could be an indicator for the presence of relevant samples in neighboring shots. As Figure 6.3 (b) shows, the related and relevant samples of the query "just one person getting out of or getting into a vehicle" are temporally neighboring.

6.3.2 Visual-based Ranking Model

6.3.2.1 Formulation

In iteration t, we aim at learning a visual-based ranking function $f^t(\mathbf{x})$ from the labeled samples $\{\mathcal{L}^t, \mathcal{Y}^t\}$. We here employ the incremental learning technique to speed up the learning process. The ranking function is formulated as a combination of three parts: (a) the ensemble of concept classifiers $\{f_k(\mathbf{x})\}_{k=1}^{K}$ related to the query; (b) the accumulation of "local ranking function" $\{\Delta f^l(\mathbf{x})\}_{l=1}^{t-1}$ learned in the previous iterations; and (c) the to-be-learned "local ranking function"

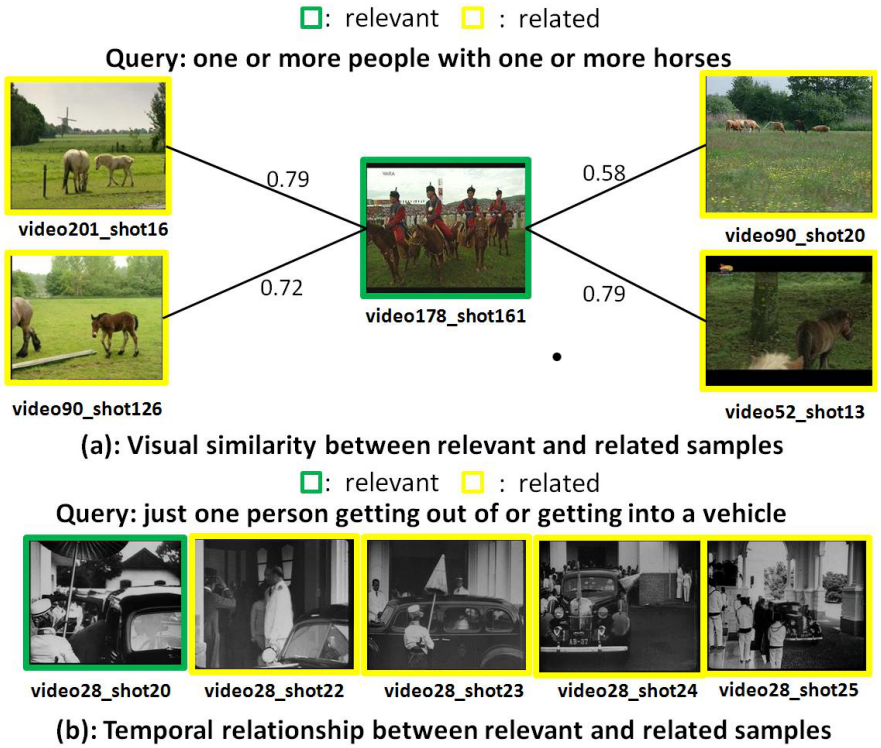


Figure 6.3: Illustration the relationship between relevant (green rectangle) and related (yellow rectangle) samples. In subfigure (a), the relevant and related samples are visually similar where the numbers on the edges represent the similarities measured by cosine distance on Color Correlogram feature. In subfigure (b), the relevant and related samples are temporally neighboring in a video.

$\Delta f^t(\mathbf{x})$.

$$f^t(\mathbf{x}) = \eta^t \sum_{k=1}^{K} d_k^t f_k(\mathbf{x}) + \frac{1}{t-1} \sum_{l=1}^{t-1} \beta_l^t \Delta f^l(\mathbf{x}) + \Delta f^t(\mathbf{x}) \tag{6.1}$$

where $\{d_k^t\}_{k=1}^K$ are the weights of concept classifiers in iteration t; $\{\beta_l^t\}_{l=1}^{t-1}$ are the weights of the previous local ranking functions; and η^t is a trade-off parameter which balances the concept classifiers and the local ranking functions. Based on above formulation, we can update the visual-based ranking function efficiently. In each iteration t, only the new labeled samples are required to learn the local ranking function $\Delta f^t(\mathbf{x})$, and the labeled samples in the previous iterations are utilized through the local ranking functions $\{\Delta f^l(\mathbf{x})\}_{l=1}^{t-1}$, which are already learned in the previous iterations.

We formulate each local ranking function as a linear discriminant function based on "*kernel trick*": $\{\Delta f^l(\mathbf{x}) = \mathbf{w}_l^T \phi(\mathbf{x})\}_{l=1}^t$, where $\{\mathbf{w}_l\}_{l=1}^t$ are model parameters and $\phi(\cdot)$ is a mapping function that maps the samples from the original space into a higher or even infinite dimensional space. As a result, $f^t(\mathbf{x})$ can be expressed as

$$f^t(\mathbf{x}) = \eta^t \sum_{k=1}^{K} d_k^t f_k(\mathbf{x}) + \frac{1}{t-1} \sum_{l=1}^{t-1} \beta_l^t \mathbf{w}_l^T \phi(\mathbf{x}) + \mathbf{w}_t^T \phi(\mathbf{x}) \tag{6.2}$$

Next we update the concept weights $\{d_k^t\}_{k=1}^K$, estimate the relatedness strength of related samples, and learn the ranking function $f^t(\mathbf{x})$,

6.3.2.2 Concept Weight Updating

The initial concept weights $\{d_k^0\}_{k=1}^K$ are obtained according to the text matching scores between the concepts and the query (see Section 6.2). However, text-based weights may not well characterize the utilities of the concept classifiers. To derive optimal ensemble of concept classifiers, we propose to optimize concept weights $\{d_k^t\}_{k=1}^K$ from users' feedbacks in iteration t. Given a labeled sample set $\mathcal{L} = \mathcal{L}_p \cup \mathcal{L}_r \cup \mathcal{L}_n$ where $\mathcal{L}_p, \mathcal{L}_r, \mathcal{L}_n$ are relevant, related and irrelevant sample sets, our basic idea is that the concept weights should make the fusion score of each relevant sample as large as possible, while ensure that of each irrelevant sample as small as possible. For each related sample, its fusion score is expected to be

Figure 6.4: Illustration of samples with different relatedness strengths.

between the average scores of relevant and irrelevant samples. Moreover, to avoid drastic fluctuation on concept weights, we make the new weight d_k^t as stable as possible (i.e., approaching to the old weight d_k^{t-1}). The new weights $\{d_k^t\}_{k=1}^K$ are optimized as follows:

$$\begin{aligned} &\arg\min_{\mathbf{d}^t} \tfrac{1}{2}||\mathbf{d}^t - \mathbf{d}^{t-1}||^2 + C(||\mathbf{I}_p - \mathbf{f}_p\mathbf{d}^t||^2 + ||\mathbf{f}_n\mathbf{d}^t||^2 \\ &+||\mathbf{f}_r\mathbf{d}^t - \tfrac{1}{|\mathcal{L}_p|}\mathbf{I}_p^T\mathbf{f}_p\mathbf{d}^t\mathbf{I}_r||^2 + ||\mathbf{f}_r\mathbf{d}^t - \tfrac{1}{|\mathcal{L}_n|}\mathbf{I}_n^T\mathbf{f}_n\mathbf{d}^t\mathbf{I}_r||^2) \\ &s.t.\ \ \mathbf{I}_k^T\mathbf{d}^t = 1, 0 \le d_k^t \le 1, k = 1, 2...K \end{aligned} \tag{6.3}$$

where $\mathbf{d}^t = [d_1^t, d_2^t, \ldots, d_K^t]^T$, $\mathbf{d}^{t-1} = [d_1^{t-1}, d_2^{t-1}, \ldots, d_K^{t-1}]^T$; $\mathbf{I}_p, \mathbf{I}_r, \mathbf{I}_n$ are column vectors with all the elements 1, the corresponding element numbers in them are $|\mathcal{L}_p|$, $|\mathcal{L}_r|$, $|\mathcal{L}_n|$ respectively; $\mathbf{f}_p$ is a $|\mathcal{L}_p| \times K$ matrix, and the i-th row, j-th column element $\mathbf{f}_p(i,j) = f_j(\mathbf{x}_i)$, the confidence score of the i-th relevant sample in $\mathcal{L}_p$ containing the concept j; Similarly, $\mathbf{f}_r$, $\mathbf{f}_n$ are $|\mathcal{L}_r| \times K$ and $|\mathcal{L}_n| \times K$ matrixes respectively. The first regularization term $||\mathbf{d}^t - \mathbf{d}^{t-1}||^2$ is employed to avoid the drastic fluctuation of the weights, the second (third) term makes the fusion score of each relevant (irrelevant) sample approach to 1 (0), and the last two terms are used to make the fusion score of each related sample between the average scores of relevant and irrelevant samples. C is a trade-off parameter. The constraint is used to normalize the new learned weight vector $\mathbf{d}^t$. This optimization problem in Eq. (6.3) can be solved using SMO algorithm [BLJ04].

6.3.2.3 Relatedness Strength Estimation

For a related sample $\mathbf{x}_i$, the relatedness strength r_i refers to the relatedness degree of $\mathbf{x}_i$ with respect to the query Q. For example, in Figure 6.4, the third sample is more related to the query "car at night street" than the second one, since it contains both concepts "car" and "street", while the second one only satisfies "street". It is impractical to ask users to label the relatedness strength, since it puts heavy burden on users. Here, we define multiple relatedness strength levels, and automatically infer the strength level of each related sample. The basic idea is that the relatedness strength of each related sample is reflected by its fusion score from related concepts. A sample that has a large fusion score is usually highly related to the query, and vice versa. Algorithm 2 describes the process of relatedness strength estimation. For each related sample $\mathbf{x}_i$, the fusion score $S(\mathbf{x}_i)$ is computed based on related concept classifiers (see step 5). With this score, the relatedness strength r_i of $\mathbf{x}_i$ is calculated as in step 6. In step 7, we obtain the relatedness strength interval $[R_k, R_{k+1}]$ for $\mathbf{x}_i$, as well as set the corresponding boundaries as $L_i = R_k$ and $U_i = R_{k+1}$. The time complexity of algorithm 2 is $\mathrm{O}(K|\mathcal{L}_r|)$.

Algorithm 2 The Relatedness Strength Estimation Algorithm

1: **Input:** The related sample set $\mathcal{L}_r$; the concept weights $\{d_k^t\}_{k=1}^K$; the concept confidence scores $\{f_k(\mathbf{x}_i)\}_{k=1}^K$ ($\mathbf{x}_i \in \mathcal{L}_r$); and N_r pre-defined relatedness strength intervals $\{[R_0,R_1], [R_1,R_2], \cdots, [R_{N_r-1},R_{N_r}]\}$, where $R_0 = -1$ and $R_{N_r} = 0$.
2: **Output:** The relatedness boundary set of related samples $\mathcal{B}=\{(L_i, U_i)\}_{i=1}^{|\mathcal{L}_r|}$.
3: **Process:**
4: **for** i=1 to $|\mathcal{L}_r|$ **do**
5: Compute $S(\mathbf{x}_i) = \sum_{k=1}^{K} d_k^t f_k(\mathbf{x}_i)$;
6: Compute $r_i = S(\mathbf{x}_i) - 1$;
7: Set $L_i = R_k, U_i = R_{k+1}$, if $R_k < r_i < R_{k+1}$, $0 \leq k \leq N_r - 1$;
8: **end for**
9: return $\mathcal{B}=\{(L_i, U_i)\}_{i=1}^{|\mathcal{L}_r|}$

6.3.2.4 Visual-based Ranking Model Learning

As compared to the typical ranking model learned on relevant and irrelevant samples, our ranking model simultaneously exploits relevant, related, and irrel-

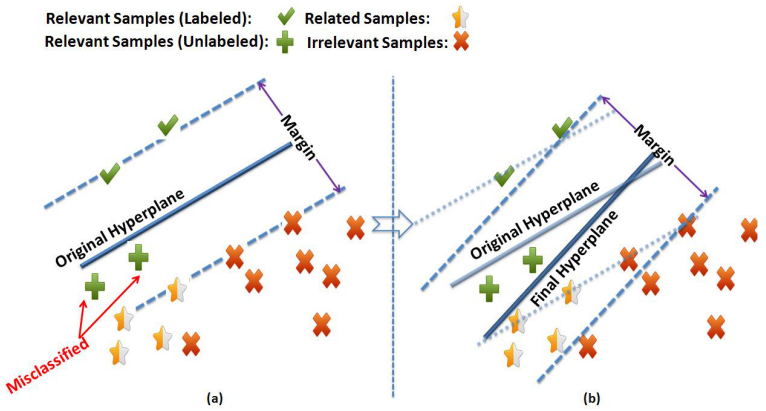


Figure 6.5: The hyperplane refinement inspired by related samples

evant samples. As shown in Figure 6.5, the related samples are located within the margin area between the hyperplane and irrelevant samples. Our basic idea is that (1) the related samples are not fully relevant to the query but closer to the classification hyperplane compared to the irrelevant samples; and (2) the related samples with larger relatedness strengths are closer to the hyperplane, and vice versa. Consequently, we derive the ranking function $f^t(\mathbf{x})$ by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}_t,\eta^t,\boldsymbol{\beta}^t} \tfrac{1}{2}||\mathbf{w}_t||^2 + C\sum_{i=1}^{N_t}(\xi_i+\zeta_i) + \tfrac{1}{2}(\eta^t-\eta^{t-1})^2 + \tfrac{1}{2}||\boldsymbol{\beta}^t-\boldsymbol{\beta}^{t-1}||^2 \\ s.t. \quad f^t(\mathbf{x}_i) \geq 1-\xi_i \quad if \;\; y_i == 1 \\ f^t(\mathbf{x}_i) \leq -1+\zeta_i \quad if \;\; y_i == -1 \\ L_i - \xi_i \leq f^t(\mathbf{x}_i) \leq U_i + \zeta_i \quad if \;\; -1 < y_i < 1 \\ \xi_i, \zeta_i \geq 0, \quad i = 1, 2, \ldots, N_t \end{aligned} \tag{6.4}$$

where $\boldsymbol{\beta}^t = [\beta_1^t, \beta_2^t, \ldots, \beta_{t-2}^t, \beta_{t-1}^t]^T$, $\boldsymbol{\beta}^{t-1} = [\beta_1^{t-1}, \beta_2^{t-1}, \ldots, \beta_{t-2}^{t-1}, 1]^T$, β_l^t is the weight for the l-th local ranking function in iteration t, ξ_i and ζ_i are slack variables, and C is the balance weight. The first term is a regularization term, which controls the model complexity. The second term is a hinge loss function, which measures the prediction error on training samples. The last two regularization terms are utilized to avoid dramatic fluctuation of parameters in successive feedback iterations. For related samples, those with strong relatedness strengths (i.e.,

large U_i and L_i) will have large output scores by the ranking function, and vice versa. η^t is the weight for the concept fusion term. The large value of η^t reflects the high utility of the concept fusion method to find relevant samples. Therefore, we determine the initial value η^0 according to the number of relevant samples contained in the search results of automatic search. We adopt a linear function $\eta^0 = 1 + \frac{|\mathcal{L}_p|}{|\mathcal{L}^0|}$ to calculate η^0, where $\mathcal{L}^0$ is the labeled sample set after automatic search, and $\mathcal{L}_p$ is the labeled relevant sample set.

To simplify the constraints in Eq. (6.4), we set the boundary $L_i = 1, U_i = +\infty$ for a relevant sample, and $L_i = -\infty, U_i = -1$ for an irrelevant sample. Thus, Eq. (6.4) can be re-written as follows:

$$\begin{aligned} \min_{\mathbf{w}_t, \eta^t, \boldsymbol{\beta}^t} \ & \tfrac{1}{2}||\mathbf{w}_t||^2 + C\sum_{i=1}^{N_t}(\xi_i + \zeta_i) + \tfrac{1}{2}(\eta^t - \eta^{t-1})^2 + \tfrac{1}{2}||\boldsymbol{\beta}^t - \boldsymbol{\beta}^{t-1}||^2 \\ s.t. \quad & f^t(\mathbf{x}_i) \geq L_i - \xi_i \quad if \ \ y_i > -1 \\ & f^t(\mathbf{x}_i) \leq U_i + \zeta_i \quad if \ \ y_i < 1 \\ & \xi_i, \zeta_i \geq 0, \quad i = 1, 2, \ldots, N_t \end{aligned} \tag{6.5}$$

The corresponding (primal) Lagrangian function is obtained as:

$$\begin{aligned} L_P = & \tfrac{1}{2}||\mathbf{w}_t||^2 + C\sum_{i=1}^{N_t}(\xi_i + \zeta_i) + \tfrac{1}{2}(\eta^t - \eta^{t-1})^2 + \tfrac{1}{2}||\boldsymbol{\beta}^t - \boldsymbol{\beta}^{t-1}||^2 - \sum_{i=1}^{N_t}\mu_i(\xi_i + \zeta_i) \\ & - \sum_{i=1}^{N_t}\alpha_i(f^t(\mathbf{x}_i) - L_i + \zeta_i) + \sum_{i=1}^{N_t}\alpha_i^{'}(f^t(\mathbf{x}_i) - U_i - \xi_i) \end{aligned} \tag{6.6}$$

where $\mu_i \geq 0$, $\alpha_i, \alpha_i^{'} \geq 0$ are Lagrange multipliers. We minimize L_P by setting its derivative with respect to $\mathbf{w}_t, \eta^t, \beta_l^t, \xi_i, \zeta_i$ to zero, which results in:

$$\begin{aligned} \mathbf{w}_t &= \sum_{i=1}^{N_t}\alpha_i\phi(\mathbf{x}_i) - \sum_{i=1}^{N_t}\alpha_i^{'}\phi(\mathbf{x}_i) \\ \eta^t &= \eta^{t-1} + \sum_{i=1}^{N_t}\alpha_i\sum_{k=1}^{K}d_k^t f_k(\mathbf{x}_i) - \sum_{i=1}^{N_t}\alpha_i^{'}\sum_{k=1}^{K}d_k^t f_k(\mathbf{x}_i) \\ \beta_l^t &= \beta_l^{t-1} + \sum_{i=1}^{N_t}\alpha_i\mathbf{w}_l^T\phi(\mathbf{x}_i) - \sum_{i=1}^{N_t}\alpha_i^{'}\mathbf{w}_l^T\phi(\mathbf{x}_i) \\ \xi_i &= C - \mu_i - \alpha_i \\ \zeta_i &= C - \mu_i + \alpha_i^{'} \end{aligned} \tag{6.7}$$

Substituting Eq. (6.7) into Eq. (6.6), we get the Lagrange dual function which is expressed in matrix form as:

$$W(\alpha_i, \alpha_i^{'}) = \arg\max_{\mathbf{A}_1,\mathbf{A}_2}[-\frac{1}{2}[\mathbf{A}_1\mathbf{A}_2]^T\mathbf{Z}[\mathbf{A}_1\mathbf{A}_2] + \mathbf{P}_1^T\mathbf{A}_1 - \mathbf{P}_2^T\mathbf{A}_2] \tag{6.8}$$

$$s.t. \qquad \mathbf{I}_{N_t}^T(\mathbf{A}_1 - \mathbf{A}_2) = 0$$

$$\begin{gathered}
where \quad \mathbf{A}_1 = [\alpha_1, \alpha_2, \ldots, \alpha_{N_t}]^T, \mathbf{A}_2 = [\alpha_1^{'}, \alpha_2^{'}, \ldots, \alpha_{N_t}^{'}]^T, 0 \leq \alpha_i, \alpha_i^{'} \leq C \\
\mathbf{P}_1 = [p_{11}, p_{21}, \ldots, p_{N_t 1}]^T, \mathbf{P}_2 = [p_{12}, p_{22}, \ldots, p_{N_i 2}]^T \\
p_{i1} = L_i - \eta^{t-1}\sum_{k=1}^{K} d_k^t f_k(\mathbf{x}_i) - \tfrac{1}{t-1}\sum_{l=1}^{t-1}\beta_l^{t-1}\mathbf{w}_l^T\phi(\mathbf{x}_i) \\
p_{i2} = U_i - \eta^{t-1}\sum_{k=1}^{K} d_k^t f_k(\mathbf{x}_i) - \tfrac{1}{t-1}\sum_{l=1}^{t-1}\beta_l^{t-1}\mathbf{w}_l^T\phi(\mathbf{x}_i) \\
\mathbf{Z} = \begin{bmatrix} \mathbf{S} & -\mathbf{S} \\ -\mathbf{S} & \mathbf{S} \end{bmatrix} \\
\mathbf{S} = (s_{ij})_{i,j=1}^{N_t}, s_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j) + \sum_{k=1}^{K} d_k^t f_k(\mathbf{x}_i) \sum_{k=1}^{K} d_k^t f_k(\mathbf{x}_j) + \tfrac{1}{t-1}\sum_{l=1}^{t-1}\mathbf{w}_l^T\phi(\mathbf{x}_i)\mathbf{w}_l^T\phi(\mathbf{x}_j)
\end{gathered}$$

Compared to the solution of the typical SVM, Eq. (6.8) involves new coefficients $\sum_{k=1}^{K} d_k^t f_k(\mathbf{x}_i) \sum_{k=1}^{K} d_k^t f_k(\mathbf{x}_j)$ and $\sum_{l=1}^{t-1}\mathbf{w}_l^T\phi(\mathbf{x}_i)\mathbf{w}_l^T\phi(\mathbf{x}_j)$ into the quadratic variables s_{ij}. This indicates that our ranking function is optimized by simultaneously taking advantages of the kernel matrix, the related concept classifiers, as well as the local ranking functions learned in the previous iterations. By solving Eq. (6.8), we obtain the model parameters $\{\alpha_i, \alpha_i^{'}\}_{i=1}^{N_t}$, which are in turn used to compute $\mathbf{w}_t$, η^t, and $\{\beta_l^t\}_{l=1}^{t-1}$ according to Eq. (6.7). Finally, the visual-based ranking function $f^t(\mathbf{x})$ is obtained according to Eq. (6.1). Given a test sample $\mathbf{x}_i$, we compute its relevance score $r_v^t(\mathbf{x}_i)$ by the sigmod function:

$$r_v^t(\mathbf{x}_i) = \frac{1}{1 + e^{-f^t(\mathbf{x}_i)}} \tag{6.9}$$

6.3.3 Temporal-based Ranking Model

Aforementioned, the relevant and related samples are usually temporally correlated. That is to say, the occurrence of a related or relevant sample usually implies the appearance of other relevant samples nearby in a video. To explore such temporal relationship, we here learn a temporal-based ranking model based graph-based semi-supervised learning technique [Zhu05].

In iteration t, we denote $\mathcal{X} = \{\mathcal{L}^t, \mathcal{U}^t\}$ as a set of N samples, where the l samples in $\mathcal{L}^t$ are labeled, and the remaining samples in $\mathcal{U}^t$ are unlabeled. We use $\mathbf{z}_\mathcal{L} = \{z_1, z_2, \ldots, z_l\}$ to represent the relevant probability of the labeled samples, where $z_i = 1$ for a relevant sample, $z_i = 0$ for an irrelevant sample, and $z_i = \sum_{k=1}^{K} d_k^t f_k(\mathbf{x}_i)$ for a related sample. An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed to model the temporal relationship between samples. The vertex set $\mathcal{V}$ corresponds to the N samples in $\mathcal{X}$, and the edge set $\mathcal{E}$ is weighted by a $N \times N$ pairwise similarity matrix $\mathbf{W} = \{W_{ij}\}_{i,j}^N$ where W_{ij} is measured as

$$W_{ij} = exp(-\frac{dis(I(\mathbf{x}_i), I(\mathbf{x}_j))^2}{\sigma^2}) \tag{6.10}$$

where $I(\mathbf{x}_i)$ is the position of a sample in a video, $dis(.,.)$ is the L_1 distance, and σ is the scaling parameter which is empirically set in experiments. When $\mathbf{x}_i$ and $\mathbf{x}_j$ are in a same video, $dis(I(\mathbf{x}_i), I(\mathbf{x}_j))$ is equal to the number of interval shots between them, otherwise, $dis(I(\mathbf{x}_i), I(\mathbf{x}_j))$ is infinite (W_{ij}=0). Based on graph $\mathcal{G}$, we next infer the relevance scores of unlabeled samples.

According to the theory of graph-based semi-supervised learning [Zhu05], a real-valued function $\mathbf{f}$ is defined to determine the relevance scores of unlabeled samples. It can be learned as follows:

$$\begin{aligned} f^* &= \arg\min_f \sum_{i,j} \frac{W_{ij}}{D_{ii}} (f_i - f_j)^2 \\ s.t. \quad & f_i \equiv z_i \quad (1 \le i \le l) \end{aligned} \tag{6.11}$$

where $D_{ii} = \sum_j W_{ij}$. Let $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$, and vector $\mathbf{f} = [\mathbf{f}_\mathcal{L}^T, \mathbf{f}_\mathcal{U}^T]^T$ denote the relevance score of all the samples. Eq. (6.11) can be transformed to its matrix form as:

$$\begin{aligned} \mathbf{f}^* &= \arg\min_\mathbf{f} \{\mathbf{f}^T(\mathbf{I} - \mathbf{P})\mathbf{f}\} \\ s.t. \quad & \mathbf{f}_\mathcal{L} \equiv \mathbf{z}_\mathcal{L} \end{aligned} \tag{6.12}$$

Split the matrix $\mathbf{P}$ after the $l-$th row and $l-$th column, we have

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{L}\mathcal{L}} & \mathbf{P}_{\mathcal{L}\mathcal{U}} \\ \mathbf{P}_{\mathcal{U}\mathcal{L}} & \mathbf{P}_{\mathcal{U}\mathcal{U}} \end{bmatrix} \tag{6.13}$$

Substitute the $\mathbf{P}$ in Eq. (6.12) with Eq. (6.13), substitute the $\mathbf{f}$ with $[\mathbf{f}_{\mathcal{L}}^T, \mathbf{f}_{\mathcal{U}}^T]^T$, and solve the resultant equations, we can obtain the temporal-based ranking function g^t as:

$$g^t: \quad \mathbf{f}_{\mathcal{U}} = (\mathbf{I} - \mathbf{P}_{\mathcal{U}\mathcal{U}})^{-1}\mathbf{P}_{\mathcal{U}\mathcal{L}}\mathbf{z}_{\mathcal{L}} \tag{6.14}$$

6.3.4 Adaptive Result Fusion

Based on the visual-based ranking model f^t and temporal-based ranking model g^t learned in each iteration t, the relevance score $r(\mathbf{x}_i)$ of sample $\mathbf{x}_i$ is generated by fusing its scores from f^t and g^t:

$$r(\mathbf{x}_i) = \lambda^t g^t(\mathbf{x}_i) + (1 - \lambda^t) r_v^t(\mathbf{x}_i) \tag{6.15}$$

where $\lambda^t \in [0, 1]$ is a balance weight in iteration t.

In order to optimally explore these two ranking models, we next propose an adaptive fusion method to automatically optimize the weight λ^t in each iteration. The basic idea is that λ^t should make the relevance scores of relevant samples as large as possible, and make that of irrelevant samples as small as possible. Meanwhile, λ^t is expected to approach the previous value λ^{t-1}. We optimize λ^t as:

$$\arg\min_{\lambda^t} \frac{1}{2}(\lambda^t - \lambda^{t-1})^2 + \frac{C_p}{|\mathcal{L}_p^t|} \sum_{\mathbf{x}_i \in \mathcal{L}_p^t} (r(\mathbf{x}_i) - 1)^2 + \frac{C_n}{|\mathcal{L}_n^t|} \sum_{\mathbf{x}_i \in \mathcal{L}_n^t} r^2(\mathbf{x}_i) \tag{6.16}$$

where $\mathcal{L}_p^t$, $\mathcal{L}_n^t$ are the labeled relevant and irrelevant sample sets in iteration t respectively, and C_p, C_n are trade-off weights. By setting the derivation of the

objective function with respect to λ^t to zero, we get the optimal solution as:

$$\begin{aligned}\lambda^t = \tfrac{1}{z}[\lambda^{t-1} + \tfrac{2C_n}{\mathcal{L}_n^t} \sum_{\mathbf{x}_i \in \mathcal{L}_n^t} (g^t(\mathbf{x}_i) r_v^t(\mathbf{x}_i) + (r_v^t(\mathbf{x}_i))^2) \\ + \tfrac{2C_p}{\mathcal{L}_p^t} \sum_{\mathbf{x}_i \in \mathcal{L}_p^t} (g^t(\mathbf{x}_i) r_v^t(\mathbf{x}_i) + (r_v^t(\mathbf{x}_i))^2 + g^t(\mathbf{x}_i) + r_v^t(\mathbf{x}_i))] \\ where: \\ z = 1 + \tfrac{2C_p}{\mathcal{L}_p^t} \sum_{\mathbf{x}_i \in \mathcal{L}_p^t} (g^t(\mathbf{x}_i) + r_v^t(\mathbf{x}_i))^2 + \tfrac{2C_n}{\mathcal{L}_n^t} \sum_{\mathbf{x}_i \in \mathcal{L}_n^t} (g^t(\mathbf{x}_i) + r_v^t(\mathbf{x}_i))^2\end{aligned} \tag{6.17}$$

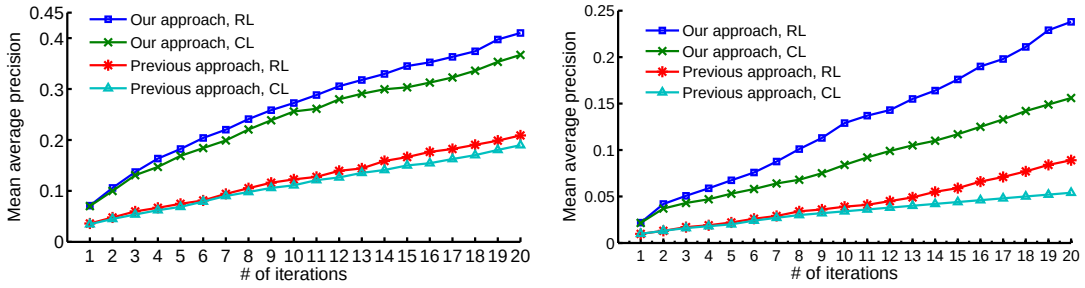
As a result, the final search results are generated by sorting the video shots according to the relevance scores in Eq. (6.15) in a descending order.

6.4 Experiments

6.4.1 Experimental Settings

We conducted experiments on two video datasets. The first dataset is the "TV08" dataset (see section 3.1.1), and the second one is the "YT11" dataset (see section 3.2.2). On "TV08" dataset, we employed Comlubia374 [YCKH07] as the primitive concept classifiers. On "YT11" dataset, the 70 concept classifiers were built using the standard Support Vector Machine (SVM) with Gaussian RBF kernel. To build the concept bundle classifiers on both datasets, we employed the SL approach in chapter 4. The parameters in SVM and SL were determined through a five-cross validation process.

To evaluate the search performance, we conducted video search on 48 queries from "TV08" dataset and 40 queries from "YT11" dataset. Given a text query, our search system first employs the automatic semantic video search approach in chapter 5 to return the initial search results. In each iteration of interactive search, user is asked to label the top 100 video shots as relevant, related, or irrelevant. Then the system updates the concept weights, learns visual-based and temporal-based ranking models. Finally, the search results are generated according to Eq. (6.15). In total, 20 feedback iterations were conducted for each query. Some parameters in our approach are set empirically. In particular, we set $C = 100$ in updating concept weights, $C = 100, \gamma = 1$(RBF kernel parameter) in



(a) "TV08" Dataset (b) "YT11" Dataset

Figure 6.6: The performance comparison in each iteration between two approaches using RL or CL measured by MAP@1000

learning the visual-based ranking function, and $\sigma = 5$ in learning the temporal-based ranking function. In the adaptive result fusion, the initial weight λ^0 in Eq. (6.15) is set as 0.5. The weight will be adjusted in each iteration according to Eq. (6.17), where the tradeoff parameters $C_p = C_n = 100$.

Average Precision (AP), which corresponds to the area under a non-interpolated recall/precision curve, was used as the performance metric. For each query, we compute the AP of the top $1,000$ search results. We then averaged the APs over all the queries, resulting in the mean average precision (MAP), which is the overall evaluation metric.

6.4.2 Evaluations

6.4.2.1 Evaluation on the Effectiveness of Related Samples

In this experiment, we investigate the utility of related samples in interactive video search. We compare two labeling strategies: Related Labeling (RL) and Conventional Labeling (CL). In RL, users are asked to label samples as relevant, irrelevant or related, while in CL users are asked to label samples as relevant or irrelevant. Based on user feedbacks, we employ two approaches to refine the search results, including the approaches proposed in our previous work [YZZ+10] and this work.

Figure 6.6 shows the performance comparison results. We can see that RL outperforms CL on both datasets, for both previous or current approaches. The performance improvement by RL is more significant on "YT11" dataset than that

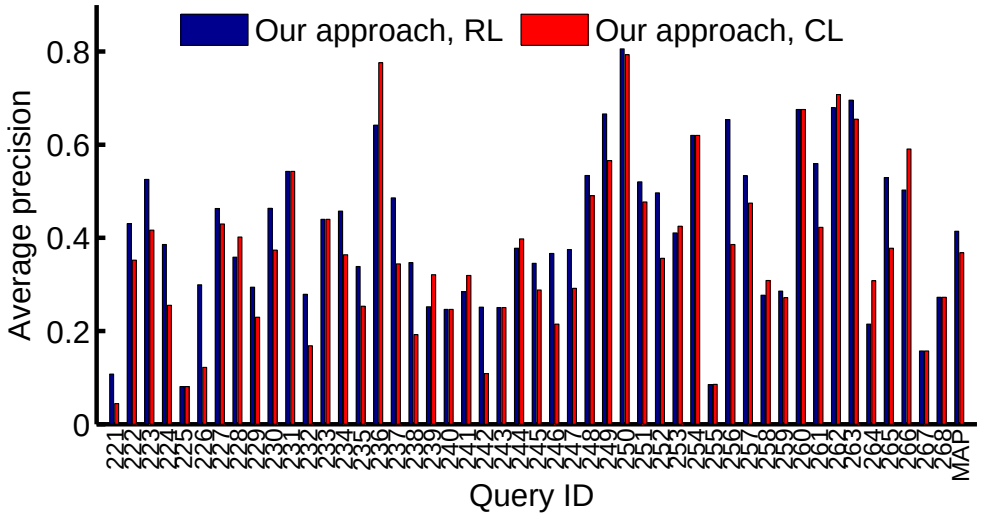


Figure 6.7: The performance of each query in the last iteration on "TV08" dataset measured by AP@1000

on "TV08" dataset. The main reason is that the relevant samples in "YT11" dataset are more sparse than those in "TV08" dataset, and the exploration of related samples can well boost search performance especially when relevant samples are sparse.

Figure 6.7 shows the APs of the 48 queries in the last iteration on "TV08" dataset. Working with our approach, RL performs better than CL on 27 queries, worse on 11 queries and the same on 10 queries. Figure 6.8 illustrates the APs of the 40 queries in the last iteration on "YT11" dataset. Compared to CL with our approach, RL with our approach performs better on 20 queries, worse on 13 queries, and the same on 7 queries.

To further analyze the utility of related samples on different queries as presented in Table 6.1 and Table 6.2, we illustrate some query attributes as well as the performance comparison. The query attributes include:

- **# of related samples**: the number of related samples labeled in the search process for a query;
- **Query Type**: a query is simple (S) or complex (C);
- **Motion**: the motion event in a query;

Table 6.1: Illustration of the query attributes on "TV08" dataset, where "RL vs. CL" means RL or CL performs better on a given query

Query ID	# of Related Samples	Query Type	Motion	RL vs. CL	Query ID	# of Related Samples	Query Type	Motion	RL vs. CL
221	14	C	open	RL	245	8	C	watch	RL
222	136	C	sit	CL	246	28	C		RL
223	37	C		RL	247	245	C		RL
224	198	C	move	RL	248	431	C		RL
225	0	S		same	249	0	S		same
226	633	C		RL	250	55	C		RL
227	46	C		RL	251	15	C	talk	RL
228	153	C	write type	CL	252	32	C	ride	RL
229	213	C		RL	253	8	C	walk	CL
230	205	C	pass	RL	254	0	C	talk	same
231	0	S		same	255	13	C	get into	RL
232	4	C	walk	RL	256	181	C	sing play	RL
233	0	C		same	257	255	C		RL
234	144	C	move	RL	258	131	C	sit	CL
235	84	C	talk	RL	259	372	C		RL
236	894	C	break	CL	260	0	S		same
237	341	C	talk	RL	261	165	C		RL
238	33	C	push	RL	262	24	C		CL
239	142	C	stand play	CL	263	233	C		RL
240	3	C		same	264	277	C		CL
241	5	C		CL	265	99	C	talk	RL
242	456	C	sit	RL	266	366	C	sit	CL
243	0	C	look	same	267	0	C	zoom	same
244	130	C	approach	CL	268	0	C		same

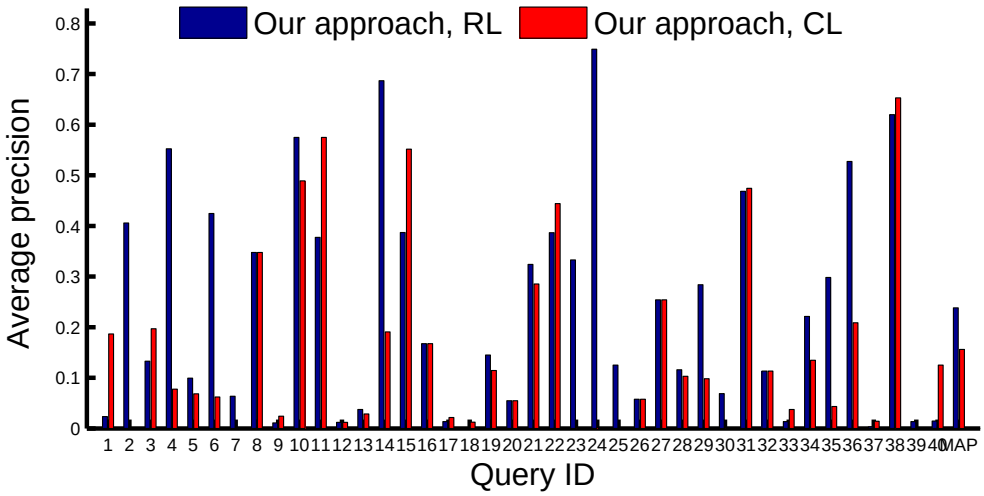


Figure 6.8: The performance of each query in the last iteration on "YT11" dataset measured by AP@1000

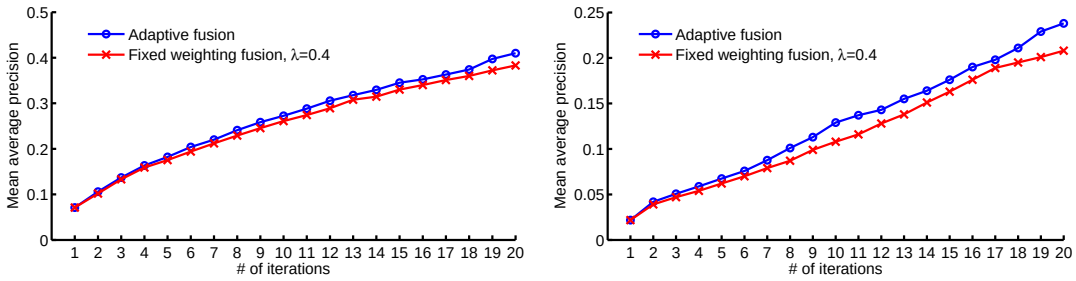
From the results in Table 6.1 and 6.2, we can derive the following observations:

1. The exploration of related samples can boost the search performance for most complex queries.

2. When few related samples are labeled for a query (e.g. the query 225, 240 in Table 6.1 and query 8, 12 in Table 6.2), RL achieves the same performance as compared to CL.

3. The related samples are ineffective in some complex queries containing motion event such as the query 228, 236, 239 (Table 6.1), and the query 3, 18, 31 (Table 6.2). This is because some motion events, such as "playing", "singing", and "writing" etc, are difficult to be modeled by the ranking functions.

The related samples may fail to find relevant samples sometimes. For example, on "TV08" dataset, for the query 266 "more than 3 people sitting at a table", user usually labels the related samples as those satisfying "3 or fewer people sitting at a table" (the query 222). Since the related and relevant samples occur mutually, the appearance of related samples actually indicates the nearby samples



(a) "TV08" Dataset (b) "YT11" Dataset

Figure 6.9: The performance comparison in each iteration between our weight updating approach and the fix weight approach measured by MAP@1000

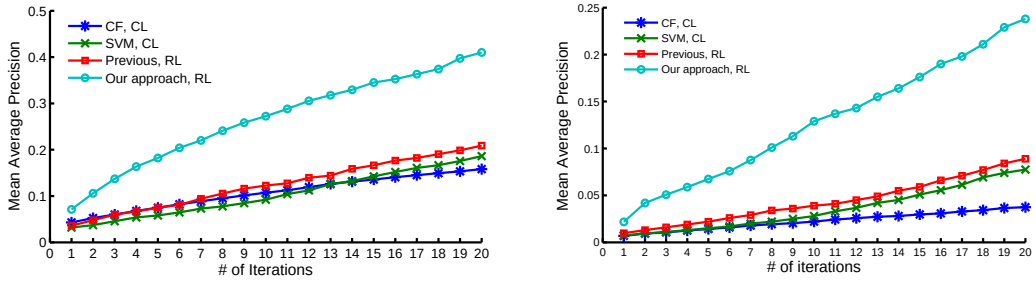
are irrelevant to the query. In such case, the temporal-based ranking function fails to predict the presence of relevant samples. The other example is the query 264 "one or more colored photographs", the related samples refer to those black and white photographs. In such case, the related and relevant samples have different visual features, and thus the visual features of related samples could decrease the search performance. In summary, the related samples are useful to find relevant samples when they are visually similar or temporally neighboring to the relevant samples.

6.4.2.2 Evaluation on Adaptive Result Fusion

This experiment investigates the effectiveness of the proposed adaptive result fusion method in Section 6.3.4. We set the initial weight λ^0 to 0.5, and adjust the value of λ^t in each iteration t according to Eq. (6.17). We compare this fusion approach to the typical fixed weighting fusion strategy, where the weight λ^t is fixed as 0.4 because this value achieves the best performance. The performance comparison is provided in Figure 6.9. We can see that the proposed adaptive fusion method performs better than the fixed weighting strategy. In particular, It obtains about 7% MAP improvement on "TV08" dataset and 14.4% MAP improvement on "YT11" dataset in the last iteration. The main reason is that the adaptive fusion approach optimizes the fusion weight based on user feedbacks and thus can optimally explore the visual-based and temporal-based ranking functions.

Table 6.2: Illustration of the query attributes on "YT11" dataset, where "RL vs. CL" means RL or CL performs better on a given query

Query ID	# of Related Samples	Query Type	Motion	RL vs. CL	Query ID	# of Related Samples	Query Type	Motion	RL vs. CL
1	77	C		CL	21	235	C	dance	RL
2	106	C		RL	22	136	C		CL
3	156	C	sing	CL	23	323	C	hunt	RL
4	65	C		RL	24	131	C	race	RL
5	615	C	cook	RL	25	42	C	see	RL
6	163	C	ride	RL	26	309	C	dance	same
7	49	C	talk	RL	27	0		eat	same
8	5	C	write	same	28	112	C	fight	RL
9	15	C		CL	29	345	C		RL
10	59	C	fly	RL	30	412	C	interview	RL
11	34	C	land	CL	31	154	C	kiss	CL
12	8	C	laugh	same	32	0		play	same
13	34	C	crash	RL	33	373	C	swim	CL
14	198	C	walk	RL	34	67	C	ride	RL
15	765	C	crash	CL	35	143	C	fight	RL
16	11	C		same	36	79	C		RL
17	76	C	dance	CL	37	274	C	crash	CL
18	45	C	watch	CL	38	551	C	fight	CL
19	168	C	fight	RL	39	354	C		RL
20	35	C		same	40	254	C	playing	CL



(a) "TV08" Dataset (b) "YT11" Dataset

Figure 6.10: The performance comparison in each iteration between our approach and the-state-of-art methods measured by MAP@1000

6.4.2.3 Comparison to the-state-of-art Methods

To demonstrate the effectiveness of our approach, we compare it against the following the-state-of-art methods:

- Support Vector Machine (SVM): In each iteration, we build an SVM classifier based on relevant and irrelevant samples. Gaussian RBF kernel is used in SVM and the parameters are set empirically ($C = 100$, $\gamma = 1$). This classifier is then used to predict the presence of the query in the unlabeled keyframes. The search results are generated according to the prediction scores.

- Concept Fusion Method (CF) [HLRYC06]: In each iteration, the search results are generated by fusing the individual results from related concepts. The initial concept weights are set according to the text matching scores [CHJ+06]. These weights are adjusted according to the approach in [HLRYC06], where a maximum posteriori probability estimation is used.

- Our Previous Approach [YZZ+10]: In each iteration, the approach first updates concept weights, then learns a visual-based ranking model based on relevant, related and irrelevant samples. The search results are generated by ordering the samples according to their relevance scores from the ranking models.

Figure 6.10 illustrates the performance comparison between our approach and the above three methods on two datasets. We can see that our approach performs the best among all the methods. It achieves a MAP of 0.41 on "TV08" dataset and 0.238 on "YT11" dataset in the last iteration. Compared to CF, SVM and our previous approach, our approach achieves a 159%, 120% and 96% performance improvement respectively on "TV08" dataset in the last iteration, while the corresponding improvements on "YT11" dataset are 526%, 205% and 167% respectively. The improvements over SVM and CF demonstrate that the related samples are beneficial to interactive video search. As aforementioned, the advantages of exploring related samples are two-fold: First, the related and relevant video segments usually share similar visual content in part due to their semantic connection, so that the related samples are beneficial to the modeling of relevant samples. Second, since video content is temporally dynamic and continuous, the occurrence of related video segments is an indicator for the presence of relevant ones in the neighboring clips. Compared to our previous approach in [YZZ+10] that exploits the visual information of related samples by the visual-based ranking model, the approach in this work further leverages the temporal relationship between the related and relevant samples by the proposed temporal-based ranking model. Through optimally exploring the visual-based and temporal-based ranking models by the adaptive fusion method, our approach achieves better performance than the previous method [YZZ+10].

6.5 Conclusion

In this chapter, we proposed to exploit "related samples" to enhance interactive semantic video search with complex queries. The "related samples" are defined as those samples that are relevant to part of the query rather than the entire query. A visual-based ranking model and a temporal-based ranking model have been developed to leverage related samples for video search. The search results are generated by fusing the results from these two models. An adaptive fusion method has been proposed to optimally explore these two models. Extensive experiments were conducted on two datasets: TRECVID 2008 and YouTube 2011 datasets. The experimental results have demonstrated the effectiveness of

the proposed approach.

Chapter 7

Application: Memory Recall based Video Search

7.1 Introduction

In our lifetime, we have seen and shot lots of photos and videos that record valuable and interesting events, places and people etc in our life. From time to time, we may want to seek a video or video segments that we have recorded or seen before for various reasons. For example, a couple may want to view their wedding video along with shots of specific friends and memorable events happened during the wedding. A girl may want to download a cartoon video she has seen in her friend's home from web. In such cases, it would be nice to provide a video search system that is able to find the desired video or video segments based on the user's memory recall. We call this "Memory Recall based Video Search" (MRVS).

To facilitate video search, the existing video search approaches focus on exploiting textual features, visual features, or semantic concepts based on users' queries. Although the state-of-art approaches have achieved some successes, they are usually ineffective when users specify a complex, inaccurate and/or incomplete query. In MRVS task, inaccurate or incomplete queries are common since people's memory recalls are usually vague, especially when the desired scenes to be recalled occurred a long time ago. This vagueness makes the state-of-art video search approaches ineffective for MRVS tasks.

Recently, a new video search task named "Known-item Search" (KIS) has emerged in TRECVID 2010 [TRE]. It aims to find a desired video that has been seen or known before by a user. In this task, a user inputs a text description of the search task, and the system returns a ranked list of results with the expectation that the correct match is ranked as high as possible. Although research on KIS task is just beginning, researchers have discovered that text-based video search is the only effective mean to tackle this problem [CWZea10; CYNea10]. MRVS is similar to KIS but with one big difference: MRVS deals more with users' personal media depositories where metadata and text descriptions are sparse, and visual matching of the desired content is often the primary mode of search. Hence the text-based techniques developed in KIS and earlier multimedia question answering approaches [NWZ+11] will not be effective. In particular, there are four challenges when applying the text-based video search approaches to MRVS tasks. First, the text words associated with the desired video are often incomplete and vague. Second, a user may remember only fragments of visual contents instead of stories or the actual conversations in the desired video, hence they are not able to provide an accurate text query. Third, many of the visual scenes are hard to describe using text. Fourth, users sometimes only want to find the desired video segments inside a long video, while the text-based approach is unable to support this because of the absence of text annotations at the video segment level. Hence, in MRVS task, users will need to issue various models of queries to recall his/her memory of a desired video.

To tackle these challenges, we design a video search system that integrates text-based, content-based and semantic video search approaches. To recall the desired video or video segments, a user may input a text query, a sequence of visual queries and concept queries, or a combination of all (see Figure 7.1). A text query is used to express the story or conversation appearing in the desired video, while a sequence of visual queries and concept queries is employed to represent some fragments of the desired video segments. In particular, since the users' visual memory is usually vague, the visual query can at best be expressed in the form of a visual sketch, while the corresponding concept query contains the list of objects/items that might appear in that visual query. In addition, we organize the visual queries and concept queries according to their temporal order in the

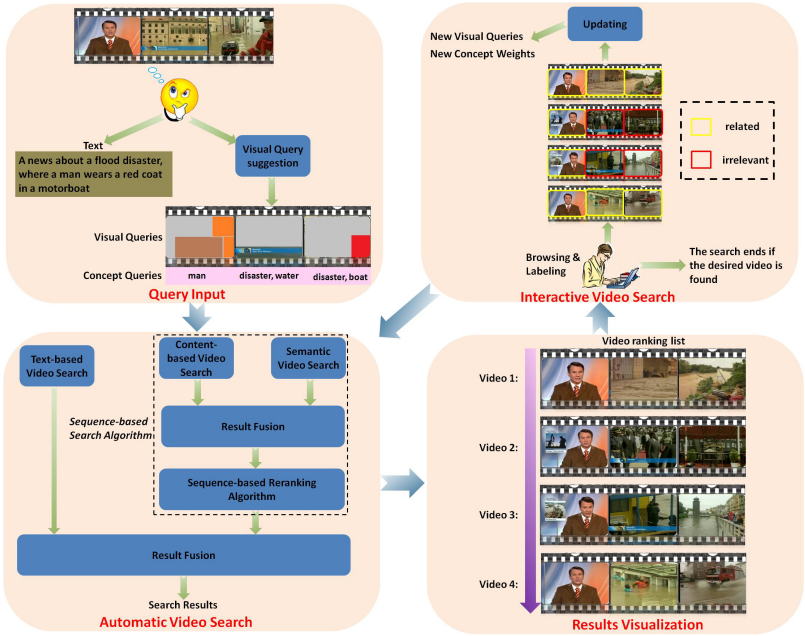


Figure 7.1: The framework of our video search system for the MRVS task

sequence. To help users in better specifying their queries, we further incorporate several functions in our system. First, for each visual query, we employ a visual query suggestion model to automatically suggest potential visual examples to help users refresh their memory. The users can then select one of the suggested visual examples to describe their desired contents more precisely. The selected visual examples are then used to compute the content-based relevance scores based on a color matching scheme. Second, in the color matching scheme, we build a color similarity matrix to allow for inexact color matching due to the fact that the users are unlikely to remember the exact colors in the desired scene. Third, for each concept query, we use the bundle-based semantic video search approach [YZZ+11a] to calculate semantic relevance scores as this approach works well for complex semantic concept inputs. Fourth, we fuse the content-based and semantic relevance scores for each video segment, and employ a reranking algorithm to generate a sequence-based relevance score by exploiting the temporal relationship among the visual and concept queries. Finally, the generated sequence-based relevance score is linearly combined with the text-based relevance score to return search results.

Furthermore, considering the fact that the query is incomplete and inexact, and there are few relevant answers, it is likely that the initial search result list may not contain the desired video. To further help the users, we incorporate the interactive video search technique for MRVS task. It consists of two serial steps: users' labeling and result updating. In the first step, as the users are unlikely to find any relevant sample[1], the system thus permits the users to label related and irrelevant samples. Related samples [YZZ+11b] are defined as those that are visually similar or semantically close to the relevant result. In the second updating step, we develop a visual query updating approach to modify the initial visual queries as well as a concept weight updating approach to adjust the concept weights. The newly generated visual queries and concept weights are then fed to the automatic video search approach to generate the new search results. We summarize our contributions as follows:

- To help users in better refreshing their memory of a desired video, we develop a visual query suggestion module to provide better visual queries, as well as a color matching scheme that allows for inexact color matching between visual queries and the desired video segments.
- We develop an algorithm to rerank the search results by exploring the temporal relationship between visual and concept queries.
- As there is often one or few relevant answers, we develop a relevance feedback scheme that allows users to label related and irrelevant samples. By exploiting visual and semantic similarity between related and relevant samples, we develop algorithms to update visual query and concept weights to refine the search results in interactive video search.

To the best of our knowledge, this is the first work that explores video search based on users' memory recalls. We conduct large-scale experiments on two video datasets: TRECVID 2010 and YouTube 2012 datasets. The experimental results demonstrate the effectiveness of our system for MRVS tasks.

[1]There is only one relevant sample for each query and the search process ends once the relevant sample is presented

7.2 Overview

7.2.1 Framework

Figure 7.1 shows the framework of the system, which consists of four parts: Query Input, Automatic Video Search, Result Visualization and Interactive Video Search. In the query input stage, a user inputs a text query Q_t, a sequence of visual and concept queries $\mathcal{Q}_s = \{Q_v^h, Q_c^h\}_{h=1}^H$, or a combination of all, based on his/her recall on the desired video. An example is shown in Figure 7.1 (see Query Input Part), where three visual queries and the corresponding concept queries are provided by users to describe the three video segments in users' memory. Based on the queries, the automatic video search generates the search results by fusing the individual results from the text-based video search and the sequence-based video search. To effectively present the search results, the system adopts a bundle-based visualization approach. Each bundle corresponds to a video result including several video segments, where the i-th segment is the search result with respect to the i-th visual and concept queries. During interactive video search, a user first labels the result samples as related or irrelevant. The system then refines the visual queries and adjusts the concept weights, which are fed to the automatic video search to generate the search results for the next iteration.

7.2.2 Visual Query Suggestion

Our system allows a user to draw a visual query Q_v^h on a sketchpad. However, the drawn visual query Q_v^h is inexact since the users' memories are usually vague. Therefore, it is desirable to automatically suggest potential visual queries based on user's rough drawing. To this end, we propose a visual query suggestion approach to tackle this problem. As Figure 7.2 shows, when a user finishes drawing a visual query Q_v^h, the system automatically suggests several potential visual query candidates. The user can then replace Q_v^h with any one of the suggested candidates if he/she thinks that the selected one is visually closer to the desired scene in the memory.

To find potential visual query candidates for Q_v^h, we calculate a visual similarity $S(Q_v^h = K_i^m | Q_t, \mathcal{Q}_s)$ between Q_v^h and each video keyframe K_i^m of the video

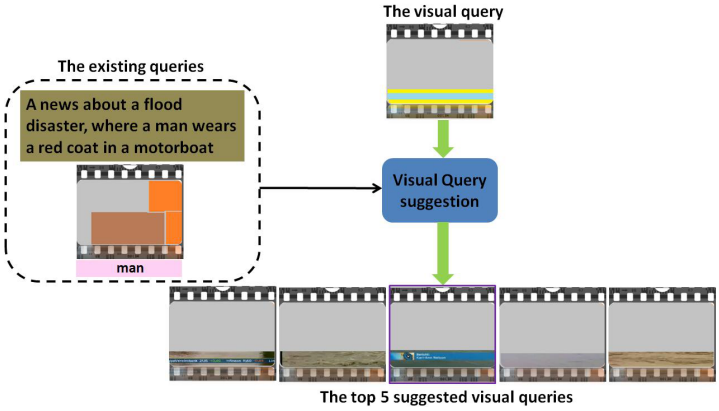


Figure 7.2: An example to illustrate visual query suggestion, where the purple rectangled visual query is selected by user to replace the drawing one.

v_m under the existing query inputs $Q_t, \mathcal{Q}_s$ below:

$$S(Q_v^h = K_i^m | Q_t, \mathcal{Q}_s) = r_v(Q_v^h, K_i^m) r_t(Q_t, d_m) r_s(\mathcal{Q}_s / \{Q_v^h, Q_c^h\}, \mathcal{K}_s / K_i^m) \quad (7.1)$$

where Q_t is the existing text query, $\mathcal{Q}_s$ is the existing sequence of visual and concept queries, d_m is the text description associated with v_m, $\mathcal{K}_s / K_i^m$ represents a keyframe sequence including neighbors of K_i^m without K_i^m, and $\mathcal{Q}_s / \{Q_v^h, Q_c^h\}$ is the sequence of visual and concept queries without $\{Q_v^h, Q_c^h\}$. $r_v(Q_v^h, K_i^m)$ calculates a content-based relevance score between Q_v^h and K_i^m (Eq. (7.2)); $r_t(Q_t, d_m)$ calculates a text-based relevance score (Section 7.3.1); and $r_s(\mathcal{Q}_s / \{Q_v^h, Q_c^h\}, \mathcal{K}_s / K_i^m)$ is a sequence-based relevance score (Eq. (7.7)). The details of these calculations are elaborated in the next section.

According to Eq. (7.1), we obtain the top K keyframes, which are the potential visual query candidates for the initial visual query Q_v^h. For each of these candidates, we only display its corresponding drawing area with respect to Q_v^h (Figure 7.2). By selecting one of these candidates, a user can replace Q_v^h with the selected one to reduce the errors in the initial visual query specification.

7.3 Automatic Video Search

7.3.1 Text-based Video Search

Given a text query Q_t, the text-based video search computes the relevance scores between Q_t and each video. Let $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ denote the text descriptions associated with the videos. The system employs the term frequency and inverse document frequency weighting scheme (*tf-idf*) [MRS09] to compute the text-based relevance scores $r_t(Q_t, d_m)$, which is widely used as a weighting factor in information retrieval and text mining.

7.3.2 Sequence-based Video Search

Besides textual terms, a user may remember certain visual scenes in the desired video or video segments. We thus prompt the user to input a visual query Q_v^h to describe some glimpses of visual content in the desired scene, and a concept query Q_c^h to specify the semantic concepts appearing in that scene. All the recalled visual scenes are organized as a sequence $\mathcal{Q}_s = \{Q_v^h, Q_c^h\}_{h=1}^{H}$, where the notation h indicates the temporal order. Based on the sequence $\mathcal{Q}_s$, the Sequence-based Video Search consists of three steps: Content-based Video Search, Semantic Video Search, and Sequence-based Reranking.

7.3.2.1 Content-based Video Search

Given a visual query Q_v^h and a keyframe K_i^m, the content-based video search estimates a relevance score by measuring their color-spatial similarity. Here, we only use colors and their rough locations instead of other visual features such as shape, texture etc. The reason is that the user is usually unable to exactly draw shapes, textures of semantic objects, while he/she can better recall color compositions of visual scenes. We divide each visual query Q_v^h and each keyframe K_i^m into 25 image blocks $\{B_j^{Q_v^h}\}_{j=1}^{25}$, $\{B_j^{K_i^m}\}_{j=1}^{25}$, where each block is represented as a 256-dimension ($16H \times 4S \times 4V$) feature vector by HSV color model [Fai05], which is a more intuitive and perceptually linear color space as compared to RGB model. The relevance score $r_v(Q_v^h, K_i^m)$ between Q_v^h and K_i^m is equal to the

weighted sum of the relevance scores from the corresponding blocks as follows:

$$r_v(Q_v^h, K_i^m) = \sum_{j=1}^{25} w(B_j^{Q_v^h}) r_v(B_j^{Q_v^h}, B_j^{K_i^m}) \tag{7.2}$$

where $r_v(B_j^{Q_v^h}, B_j^{K_i^m})$ is the relevance score between $B_j^{Q_v^h}$ and $B_j^{K_i^m}$, and $w(B_j^{Q_v^h})$ is the weight of the block $B_j^{Q_v^h}$. A block with a higher weight means that it is more important in the measurement. Here, we postulate that the importance of a block $B_j^{Q_v^h}$ is proportional to the area size painted in this block, thus we set $w(B_j^{Q_v^h})$ as the proportion of the painted area in $B_j^{Q_v^h}$ to that in Q_v^h as follows:

$$w(B_j^{Q_v^h}) = \frac{\mathbf{f}(B_j^{Q_v^h})^T \mathbf{1}}{\sum_{j'=1}^{25} \mathbf{f}(B_{j'}^{Q_v^h})^T \mathbf{1}} \tag{7.3}$$

where $\mathbf{f}(B_j^{Q_v^h})$ is a 255-dimension feature vector (excluding the one dimension which is the background color in the sketchpad), and $\mathbf{1}$ denotes a vector with all of its elements equal to one.

To calculate $r_v(B_j^{Q_v^h}, B_j^{K_i^m})$ between two blocks, one challenge is to overcome the inexact color matching problem. It is very likely that a user may use a similar color instead of the exact one in the desired scene to draw the visual query Q_v^h. To tackle this problem, we employ a perceptually linear color space HSV in which the computational difference between two colors is proportional to human perceptual difference between colors [Fai05]. Let a and b be two colors, and (h_a, s_a, v_a), (h_b, s_b, v_b) be their HSV values respectively, the color similarity $S_{a,b}$ between a and b can be calculated [1] as follows:

$$S_{a,b} = 1 - 1/\sqrt{5}[(v_a - v_b)^2 + (s_a \cos h_a - s_b \cos h_b)^2 + (s_a \sin h_a - s_b \sin h_b)^2] \tag{7.4}$$

In this way, we build a color similarity matrix $\mathbf{S}$ to account for the slight variations in color specification by the users. The relevance score $r_v(B_j^{Q_v^h}, B_j^{K_i^m})$ is computed

[1] http://www.ee.columbia.edu/ln/dvmm/researchProjects/MultimediaIndexing/VisualSEEk/acmmm96/node8.html#eqcoldist

by measuring the color similarity as follows:

$$r_v(B_j^{Q_v^h}, B_j^{K_i^m}) = \frac{\mathbf{f}(B_j^{Q_v^h})^T \mathbf{S} \mathbf{f}(B_j^{K_i^m})}{\sqrt{\mathbf{f}(B_j^{Q_v^h})^T \mathbf{S} \mathbf{f}(B_j^{Q_v^h}) \mathbf{f}(B_j^{K_i^m})^T \mathbf{S} \mathbf{f}(B_j^{K_i^m})}} \tag{7.5}$$

where the numerator calculates the color similarity by considering the correlation between different colors, and the denominator is a normalization term.

7.3.2.2 Semantic Video Search

Besides sketching a visual query, it is also convenient for a user to specify the semantic concepts appearing in the desired visual scene to form a concept query Q_c^h. One advantage of concept query is that it can add semantic information in the query input, which complements the inadequacy of visual query since it is difficult for the user to draw semantic objects.

To calculate the semantic relevance scores $r_c(Q_c^h, K_i^m)$ between Q_c^h and each keyframe K_i^m, we employ the bundle-based semantic video search approach [YZZ+11a]. This approach utilizes a high-level concept descriptor named "Concept Bundle", that integrates multiple primitive concepts and the relationship between that, to perform semantic video search. Because a recalled visual scene usually contains multiple semantic concepts, a concept bundle can better interpret the concept query Q_c^h as compared to just the primitive concepts in MRVS task. The process of the semantic video search is as follows: We first define a set of primitive concepts and build the corresponding concept classifiers. We then select a set of informative concept bundles based on the primitive concepts, and build the corresponding concept bundle classifiers. Finally, for each concept query Q_c^h, we calculate its semantic relevance scores $r_c(Q_c^h, K_i^m)$ with respect to each keyframe K_i^m. The details of the process can be found in [YZZ+11a].

7.3.2.3 Sequence-based Reranking

For each visual query Q_v^h and concept query Q_c^h, the approaches estimate a content-based relevance score $r_v(Q_v^h, K_i^m)$ and a semantic relevance score $r_c(Q_c^h, K_i^m)$. The relevance score $r_b(\{Q_v^h, Q_c^h\}, K_i^m)$ between $\{Q_v^h, Q_c^h\}$ and the keyframe K_i^m is

calculated by linearly fusing the scores $r_v(Q_v^h, K_i^m)$ and $r_c(Q_c^h, K_i^m)$ as follows:

$$r_b(\{Q_v^h, Q_c^h\}, K_i^m) = \alpha * r_v(Q_v^h, K_i^m) + (1-\alpha) * r_c(Q_c^h, K_i^m) \tag{7.6}$$

where α is a weight. We set it to 0.5 to assign equal weights to the content-based and semantic video search results.

In real cases, some users may remember several visual scenes and are able to specify the temporal order among them. Thus temporal relationship can be explored to increase the search performance. Here, we propose a reranking algorithm to explore temporal relationship.

Given the sequence $\mathcal{Q}_s = \{Q_v^h, Q_c^h\}_{h=1}^H$, where $\{Q_v^h, Q_c^h\}$ occurs before $\{Q_v^{h+1}, Q_c^{h+1}\}$, the sequence-based reranking algorithm aims to find a subset of keyframes from a continuous keyframe sequence $\mathcal{K}_s = \{K_1^m, \ldots, K_W^m\}$ to optimally match $\mathcal{Q}_s$. Here, W is the window size which we set it as the number of keyframes in a video in experiments. We transform the calculation of the sequence-based relevance score $r_s(\mathcal{Q}_s, \mathcal{K}_s)$ to select an optimal subset with H keyframes from $\mathcal{K}_s$ to best match the H queries in $\mathcal{Q}_s$. Meanwhile, this matching should be consistent in temporal order. Here, the optimal subset means that it can maximize $r_s(\mathcal{Q}_s, \mathcal{K}_s)$ among all choices. The formula is as follows:

$$\begin{aligned} & r_s(\mathcal{Q}_s, \mathcal{K}_s) = \max_{i_1, i_2, \ldots, i_H} \prod_{h=1}^{H} r_b(\{Q_v^h, Q_c^h\}, K_{i_h}^m) \\ & s.t.\ 1 \le i_1 < i_2 < \ldots < i_H \le W \end{aligned} \tag{7.7}$$

A direct and exact solution to this problem has an extremely high computational cost, which is $O(W^H)$. This time cost is too high to conduct online video search. To reduce the computational cost, we propose an approximate greedy algorithm to solve Eq. (7.7). As Algorithm 1 shows, for each $\{Q_v^h, Q_c^h\}$, the algorithm first finds a keyframe $K_{i_h}^m$ from the sequence $\mathcal{K}_s$ that maximizes the value of $r_b(\{Q_v^h, Q_c^h\}, K_{i_h}^m)$ (line 4-6). It then checks the temporal order between any two keyframes $K_{i_h}^m$ and $K_{i_{h'}}^m$. If the temporal order is wrong (line 9), the algorithm reselects one keyframe to satisfy the temporal order constraint as well as to maximize the relevance score $r_b(\{Q_v^h, Q_c^h\}, K_{i_h}^m) r_b(\{Q_v^{h'}, Q_c^{h'}\}, K_{i_{h'}}^m)$ (line 10). The approximate algorithm reduces the computational cost to $O(H^2 W)$.

Algorithm 3 Sequence-based Reranking Algorithm

Input: The keyframe sequence $\mathcal{K}_s = \{K_{i_h}^m\}_{i_h=1}^{W}$, the sequence $\mathcal{Q}_s = \{Q_v^h, Q_c^h\}_{h=1}^{H}$, the relevance scores $\{r_b(\{Q_v^h, Q_c^h\}, K_{i_h}^m)\}_{h=1,i_h=1}^{h=H,i_h=W}$ $(W \geq H)$
Output: the keyframe subset $\{K_{i_1}^m, K_{i_2}^m, \ldots, K_{i_H}^m\}$, the relevance score $r_s(\mathcal{Q}_s, \mathcal{K}_s)$
Process:
for $h = 1$ to H **do**
    $i_h = \arg \max_{1 \leq i_{h'} \leq W} r_b(\{Q_v^h, Q_c^h\}, K_{i_{h'}}^m)$
end for
for $h = 1$ to $H - 1$ **do**
    for $h' = h + 1$ to H **do**
        if $i_h \geq i_{h'}$ **then**
            Change i_h or $i_{h'}$ to satisfy $i_h < i_{h'}$ as well as maximize the value $r_b(\{Q_v^h, Q_c^h\}, K_{i_h}^m) r_b(\{Q_v^{h'}, Q_c^{h'}\}, K_{i_{h'}}^m)$
        end if
    end for
end for
$r_s(\mathcal{Q}_s, \mathcal{K}_s) = r_b(\{Q_v^1, Q_c^1\}, K_{i_1}^m) r_b(\{Q_v^2, Q_c^2\}, K_{i_2}^m) \ldots r_b(\{Q_v^H, Q_c^H\}, K_{i_H}^m)$
return $\{K_{i_1}^m, K_{i_2}^m, \ldots, K_{i_H}^m\}, r_s(\mathcal{Q}_s, \mathcal{K}_s)$

7.3.3 Visualization

Based on the results from the text-based video search and the sequence-based video search, the final search results are generated according to the fusion scores from these two parts, where the fusion weights are set empirically in our experiments.

To effectively present the search results to users, the system employs a bundle-based visualization approach as shown in Figure 7.1. Here, a bundle contains H keyframes in a video, where each keyframe matches one of $\{Q_v^h, Q_c^h\}_{h=1}^{H}$. The H keyframes in a bundle could be returned by Algorithm 1.

Our bundle-based visualization approach is a trade-off between the video-based and keyframe-based visualization approaches, which list videos or keyframes one by one in the interface. One advantage of this visualization is that the system can simultaneously present video results as well as several interesting and impressive keyframes. Moreover, these keyframes can be labeled in the interactive video search to further refine the search results.

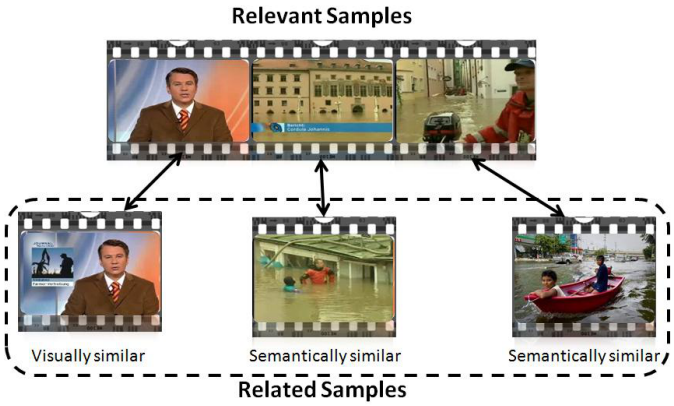


Figure 7.3: An example to illustrate related samples in MRVS task

7.4 Interactive Video Search

The interactive video search consists of two steps: 1) User' labeling; and 2) Result updating.

7.4.1 Labeling

In the traditional interactive video search, a user labels samples as relevant and irrelevant. However, in MRVS task, there is likely to be only one or few relevant answers for each query, which means that the possibility of finding a relevant sample is rare and it will end the search in most cases. Therefore, users hardly have opportunities to label relevant samples during the search process, which makes the interactive video search ineffective. Thus, we resort to related samples, which are more frequently seen in the interactive search results.

To tackle this problem, we allow users to label related and irrelevant samples. In MRVS task, we define related samples as those that are either visually similar or semantically close to the relevant samples [YZZ+11b]. Some examples of related samples are illustrated in Figure 7.3.

To determine whether a related sample is visually similar or semantically close to the relevant sample, the system first calculates a content-based relevance score between the related sample and the corresponding visual query. If this score is larger than a threshold, then this related sample is visually similar with the

relevant sample. Otherwise, it is deemed to be semantically close to the relevant sample. We use the visually similar related samples to update the visual queries, and the semantically similar related samples to update the concept weights.

7.4.2 Result Updating

Based on the labeled related and irrelevant samples, the result updating approach refines the search results from two aspects: modifying the visual queries, and adjusting the concept weights. The updated visual queries and concept weights are then fed to the automatic video search to generate new search results in the next iteration.

7.4.2.1 Adjusting the Visual Queries

The initial visual queries may be inexact, thus we need to adjust the visual queries in the interactive search process. Given a sequence of visual queries $\{Q_v^h\}_{h=1}^H$, this step aims to map the visual queries into the new ones $\{I_v^h\}_{h=1}^H$ based on the labeled related and irrelevant samples. Let $\mathcal{X}_h$ $(1 \leq h \leq H)$ be the labeled sample set with respect to Q_v^h. For each sample $x_i^h \in \mathcal{X}_h$, it can be a related sample or an irrelevant one. Our basic idea is that the new visual query I_v^h should be dissimilar to the irrelevant samples in $\mathcal{X}_h$, and similar to the related samples in $\mathcal{X}_h$. Meanwhile, we should penalize the visual difference between I_v^h and Q_v^h. We express this idea in an optimization framework as follows:

$$\begin{aligned} \min_{\mathbf{f}(B_j^{I_v^h})} & \frac{C_1}{R}\sum_{r=1}^{R}||\mathbf{f}(B_j^{I_v^h}) - \mathbf{f}(B_j^{x_r^h})||^2 - \frac{1}{N}\sum_{n=1}^{N}||\mathbf{f}(B_j^{I_v^h}) - \mathbf{f}(B_j^{x_n^h})||^2 \\ & + C_2 * (\mathbf{f}(B_j^{I_v^h}) - \mathbf{f}(B_j^{Q_v^h}))^T \mathbf{S}(\mathbf{f}(B_j^{I_v^h}) - \mathbf{f}(B_j^{Q_v^h})) \\ s.t. \quad & \mathbf{f}(B_j^{I_v^h})^T \mathbf{1} \leq 1, \mathbf{f}_d(B_j^{I_v^h}) \geq 0, d = 1, 2, \ldots, 255 \end{aligned} \tag{7.8}$$

where x_r^h, x_n^h are related and irrelevant samples, R, N are the number of x_r^h, x_n^h in $\mathcal{X}_h$ respectively, $\mathbf{S}$ is the color similarity matrix, $\mathbf{f}_d(B_j^{I_f^h})$ is the value of the feature vector $\mathbf{f}(B_j^{I_f^h})$ on dimension d, and C_1, C_2 are the weights. The first term in Eq. (7.8) makes the new visual query similar to the related samples. We always set $C_1 > 1$ to alleviate the imbalance problem between the related and irrelevant

samples. The second term ensures that the new visual query is dissimilar to the irrelevant samples. The third term is a penalty term on visual difference between Q_v^h and I_v^h. We choose this penalty term since it satisfies the following two conditions: First, if $Q_v^h = I_v^h$, the penalty cost is zero. Second, during the mapping from Q_v^h to I_v^h, the more similar color it is mapped to, the lower cost it will generate.

The optimization problem in Eq. (7.8) is an inequality constrained minimization problem. It is easy to prove that this problem is a constrained convex optimization problem, and has a global solution. We employ the Augmented Lagrangian Method [Hes69] to solve this problem.

7.4.2.2 Adjusting the Concept Weights

For each concept query Q_c^h, semantic video search calculates a semantic relevance score with respect to each keyframe candidate. In this approach, we determine a concept weight according to the classifier performance of the concept and the semantic relatedness between the concept and the query based on the text-matching scores [YZZ+11a]. This computation is not accurate because of the following two reasons: First, the evaluation of classifier performance is inaccurate. Second, the text-matching scores based on external text source cannot accurately reflect the concept distribution in the video dataset. Therefore, we need to adaptively adjust the concept weights in the interactive search process.

We propose an optimization algorithm to update the concept weights. Let the concept query Q_c^h be mapped to K related concepts $\{C_k\}_{k=1}^K$, where d_k^{t-1} is the concept weight of C_k in iteration $t-1$. In iteration t, the optimization algorithm aims to adjust d_k^{t-1} to d_k^t based on the labeled related and irrelevant sample set $\mathcal{L}_r, \mathcal{L}_n$. Our basic idea is that the concept weights should make the semantic relevance score of each related sample as large as possible, while ensuring that of each irrelevant sample to be as small as possible. Moreover, to avoid drastic fluctuation on concept weights, we make the new weight d_k^t as stable as possible. The new weights $\{d_k^t\}_{k=1}^K$ are optimized as follows:

$$\begin{aligned} \arg\min_{\mathbf{d}^t} \tfrac{1}{2}||\mathbf{d}^t - \mathbf{d}^{t-1}||^2 + C(||\mathbf{I}_r - \mathbf{R}_r\mathbf{d}^t||^2 + ||\mathbf{R}_n\mathbf{d}^t||^2) \\ s.t.\ \ \mathbf{I}_k^T\mathbf{d}^t = 1, 0 \le d_k^t \le 1, k = 1, 2...K \end{aligned} \tag{7.9}$$

where $\mathbf{d}^t = [d_1^t, d_2^t, \ldots, d_K^t]^T$, $\mathbf{d}^{t-1} = [d_1^{t-1}, d_2^{t-1}, \ldots, d_K^{t-1}]^T$; $\mathbf{I}_r, \mathbf{I}_k$ are column vectors with r, k elements equal to 1; C is a trade-off parameter; $\mathbf{R}_r$ is a $|\mathcal{L}_r| \times K$ matrix, and the i-th row, j-th column element is the semantic relevance score of the i-th related sample $\mathbf{x}_i$ in $\mathcal{L}_r$ containing the concept C_j. Similarly, $\mathbf{R}_n$ is a $|\mathcal{L}_n| \times K$ matrix. The first regularization term $||\mathbf{d}^t - \mathbf{d}^{t-1}||^2$ is employed to avoid the drastic fluctuation of the weights, the second (third) term makes the semantic relevance score of each related (irrelevant) sample approach to 1 (0). The constraint is used to normalize the weight vector $\mathbf{d}^t$. This optimization problem in Eq. (7.9) can be solved using SMO algorithm [BLJ04].

7.5 Experiments

7.5.1 Experimental Settings

We conducted experiments on two video datasets. The first one is the "TV10" dataset (see section 3.1.2), and the other one is "YT12" dataset(see section 3.2.3). To perform semantic video search, we built the classifiers of the 130 primitive concepts and the 40 informative concept bundles (see Table 7.1) selected by the approach as described in [YZZ+11a] on both datasets. In particular, we directly downloaded the 130 primitive concept classifiers from the CU-VIREO374 website[1] for "TV10" dataset, and trained that by using the LibSVM algorithm[2] for "YT12" dataset. We then trained the classifiers of the 40 concept bundles by using the multi-task SVM algorithm in chapter 4 for both datasets. All the parameters in the algorithms were set through the fivefold cross-validation process.

7.5.2 Experimental Results

7.5.2.1 Evaluation on Automatic Video Search

We perform automatic video search on the 298 queries from "TV10" dataset and 50 queries from "YT12" dataset. In particular, we set the fusion weight $\alpha = 0.5$ in Eq.(7.6) to give equal weights to the results from content-based and semantic

[1] http://www.ee.columbia.edu/ln/dvmm/CU-VIREO374/
[2] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

Table 7.1: The 40 informative concept bundles selected based on the 130 primitive concepts from TRECVID 2010 concept detection task, where we filtered the results by using WordNet to avoid the "the-kind-of" and "the-part-of" relationship between two primitive concepts in a concept bundle.

1	swimming,waterscape_waterfront	21	computer_or_television_screens,office
2	road,car	22	helicopter_hovering,sky
3	boat_ship,waterscape_waterfront	23	walking,shopping_mall
4	crowd,outdoor	24	landscape,mountain
5	plant,mountain	25	motorcycle,nighttime,racing
6	highway,car	26	celebrity_entertainment,teenagers
7	airplane,military	27	celebrity_entertainment,teenagers,singing
8	building,cityscape	28	demonstration_or_protest,explosion_Fire
9	crowd,demonstration_or_protest	29	mountain,flower
10	beach,sky	30	conference_room,meeting
11	cityscape,sky	31	natural_disaster,waterscape_waterfront
12	car,road,racing	32	sports,stadium
13	kitchen,female person	33	snow,mountain
14	beach,swimming	34	computer_or_television_screens,office,meeting
15	bicycles,road	35	people_marching,demonstration_or_protest
16	car,explosion_fire	36	weather,reportor
17	desert,sky	37	helicopter_hovering,military
18	telephones,office	38	bridges,boat_ship
19	building,sky	39	flag,military
20	motorcycle,nighttime	40	singing,celebrity_entertainment

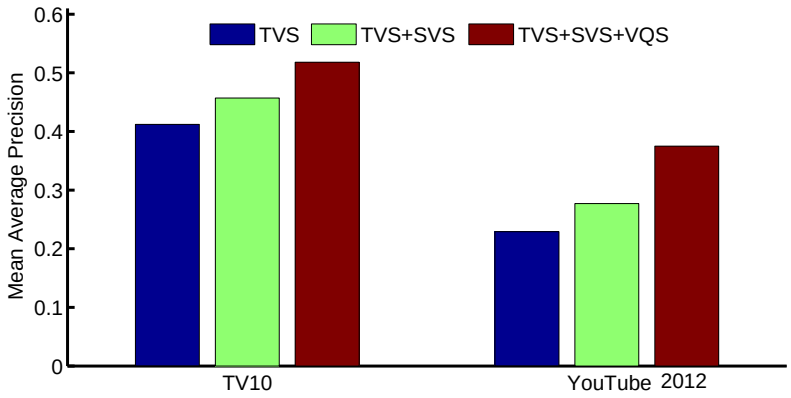


Figure 7.4: The performance comparison among the three approaches measured by MAP@100

video search approaches. The balancing weights for text-based video search and sequence-based video search was set to 0.3 and 0.7 respectively based on the performance evaluation on the 122 training queries from the TRECVID 2010 KIS task.

The search performance was measured by the Average Precision (AP). For each query, we measured the AP on the top 100 search results. All the AP values are averaged to obtain the Mean Average Precision (MAP) as a measure of the overall performance.

Experiment 1: This experiment evaluates the effectiveness of the visual and concept queries as well as that of the visual query suggestion approach. We compare the following three approaches: (1) Text-based Video Search (TVS) which returns search results based only on text queries; (2) Text-based Video Search + Sequence-based Video Search (TVS+SVS) which returns search results based on text queries and a sequence of visual and concept queries; and (3) Text-based Video Search + Sequence-based Video Search + Visual Query Suggestion (TVS+SVS+VQS) which further involves the visual query suggestion approach to provide better visual queries. Figure 7.4 shows the comparison result. On "TV10" dataset, the MAPs of the three approaches are 0.412, 0.457, and 0.518 respectively, while that for the "YT12" dataset are 0.229, 0.277, and 0.375 respectively. Compared to the pure text-based video search, TVS+SVS

approach achieves a 10.9% improvement on "TV10" dataset, and 21.0% improvement on "YT12" dataset. This verifies that the visual and concept queries are useful in MRVS tasks. Moreover, the visual query suggestion further enhances the search performance, with 13.3% and 35.4% improvements on "TV10" and "YT12" datasets, respectively. This shows that the visual query suggestion can further improve the search performance by providing better visual queries.

Figure 7.5 shows the number of queries that achieve the best performance by the three approaches (TVS, TVS+SVS, TVS+SVS+VQS) on the two video datasets. There are 203 queries on "TV10" dataset and 35 queries on "YT12" dataset that the system could find the right answers by using one of the three approaches. For the remaining queries, the system does not return the right answer within the top 100 search results. From the results, we observe that TVS performs best on 4 queries of "TV10" dataset and 0 query of "YT12" dataset, while TVS+SVS achieves the best performance on 23 queries of "TV10" dataset and 5 queries of "YT12" dataset. This result again shows that the introduction of visual and concept queries improves the search performance on most queries. However, it may worsen the performance of some queries because user may input incorrect visual and concept queries based on his/her vague memory. Finally, the use of visual query suggestion could achieve the best performance on 38 queries of "TV10" dataset and 8 queries of "YT12" dataset.

Figure 7.6 shows the automatic search results by these three approaches on Query 8. TVS does not return the right answer in the top 9 search results, while TVS+SVS that utilizes the visual and concept queries is able to find the right video but at a low rank of 8. By using visual query suggestion, TVS+SVS+VQS is able to return the right video in the top rank of the result list.

Experiment 2: This experiment evaluates the effectiveness of the Sequence-based Video Search in details based on its three subcomponents: Content-based Video Search, Semantic Video Search and Sequence-based Reranking algorithm. We present the performance comparison results in Table 7.2, and present the detailed explanation as follows:

- In content-based video search, we evaluate the effectiveness of the use of color similarity matrix **S**. Table 7.2 shows that the MAP by using **S** is 0.518 and 0.375 on "TV10" and "YT12" datasets, respectively. This per-

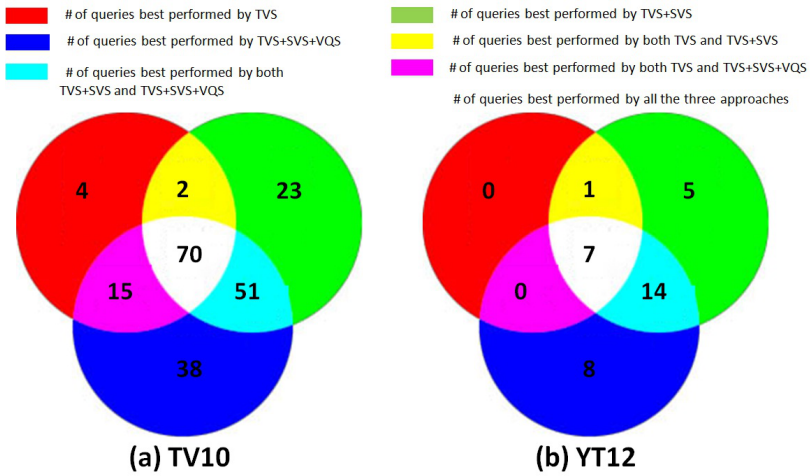


Figure 7.5: The illustration of the number of queries best performed by the three approaches

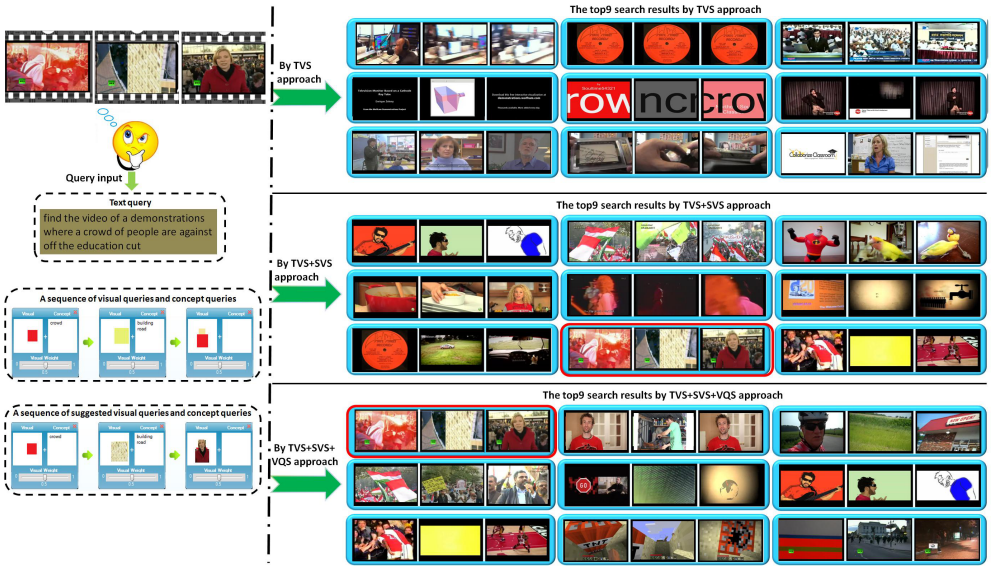


Figure 7.6: An example to compare the automatic search results from the TVS, TVS+SVS, and TVS+SVS+VQS approaches. We list the top 9 retrieved video results by these three approaches on Query 8 in the Table 3.10, where the rank lists are ordered from left to right and top to bottom (relevant samples are marked in red boxes). Each video result is represented by three inside video shots corresponding to the three visual and concept queries except for the TVS approach where the three video shots in a video result are randomly selected.

Table 7.2: Illustration of the effectiveness of SVS from the aspects of using color similarity matrix (**S**) in content-based video search, concept bundle (CB) and classifier performance (CP) in semantic video search, and temporal order (TO) in sequence-based reranking algorithm. "+"/"-" preceding the aspects (**S**, CB, CP & TO) mean the overall method incorporates or not incorporates any of these aspect. The performance is measured in terms of MAP@100.

Dataset	Content-based Video Search		Semantic Video Search				Sequence-based Reranking	
Name	+**S**	−**S**	+CB,+CP	+CB,-CP	-CB,+CP	-CB,-CP	+TO	-TO
TV10	0.518	0.467	0.518	0.495	0.483	0.466	0.549	0.516
YT12	0.375	0.342	0.375	0.348	0.341	0.325	0.448	0.418

formance is better than that from the approach without using **S**, where the corresponding MAP is 0.467 and 0.342 respectively. The reason is that general users cannot exactly draw the colors in the video segment, and thus it is helpful to bridge the color difference by using **S**.

- In semantic video search, we demonstrate the usefulness of concept bundles and concept selection strategy. We evaluate the semantic video search under four settings, where +CB/-CB means the search is performed based on both concept bundle and primitive concepts or only primitive concepts; and +CP/-CP means the concept selection strategy considers/ignores classifier performance. As can be observed, the use of concept bundles can enhance the search performance, and the incorporation of classifier performance during concept selection is also effective for performance improvement.

- In sequence-based reranking algorithm, we aim to validate the effectiveness of exploiting the temporal order between visual and concept queries. We performed the search under two settings: the sequence-based reranking algorithm that considers/ignores the temporal order (+TO/-TO). We implement -TO algorithm according to Eq. (7.7) by deleting the temporal order constraint. Since a user may input only one visual and concept query where temporal order is not applicable, we exclude those queries and only show the results for queries containing more than one visual and concept queries. This gives rise to 149 queries on "TV10" dataset and 36 queries on "YT12" dataset. By exploiting the temporal order, the sequence-based

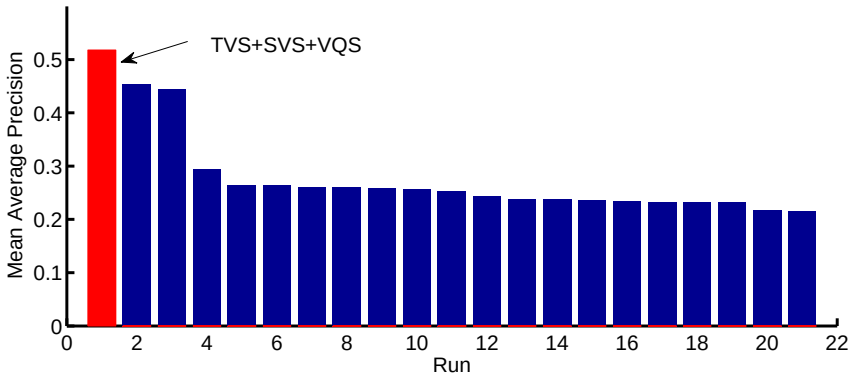


Figure 7.7: MAP comparison with the top-20 official submissions in TRECVID 2010 known-item search task

reranking algorithm achieves 6.4% and 7.2% performance gains on "TV10" and "YT12" datasets respectively. This result clearly demonstrates the effectiveness of exploiting the temporal order in improving the search performance.

Finally, Figure 7.7 compares our search results to the official submissions on "TV10" dataset in Trecvid 2010 KIS task. It shows that our approach performs the best among all the submissions.

7.5.2.2 Evaluation on Interactive Video Search

We conducted the interactive video search for MRVS tasks on 24 queries (the query 1-24) from "TV10" dataset, and 50 queries from "YT12" dataset. Given a text query as well as a sequence of visual and concept queries, the automatic video search (TVS+SVS+VQS) returns the top 100 search results, and the user (who has previously viewed the desired video with respect to this query) can optionally label the returned samples as related or irrelevant. Based on the labeled samples, the system updates the visual queries and the concept weights, which are then fed to the automatic video search to generate the new search results in the next iteration. In the experiments, we empirically set $C_1 = 3$ and $C_2 = 1$ in Eq. (7.8) for visual query updating, and $C = 100$ in Eq. (6.3) for concept weights updating. To record search results, in each two minutes of interactive search, we asked the

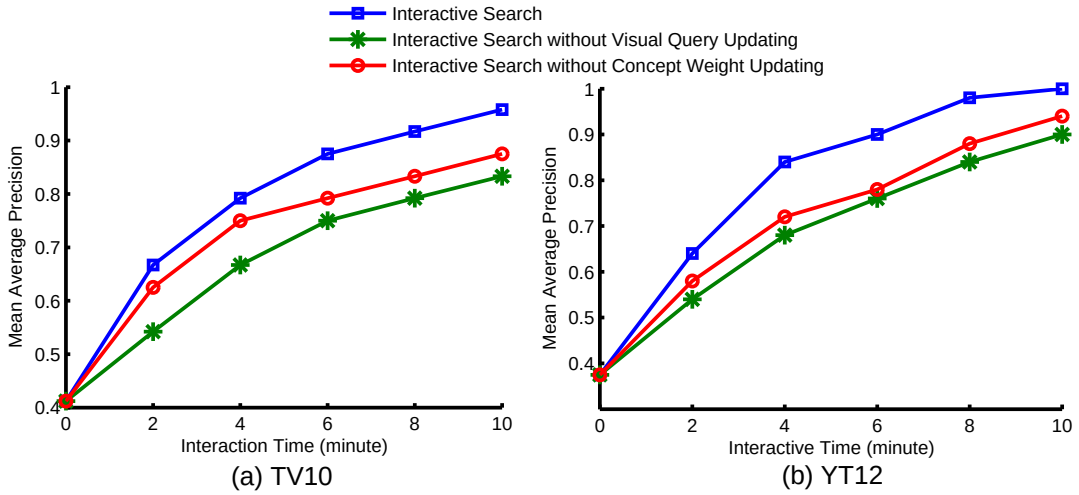


Figure 7.8: The comparison of video search performance by using or not using visual query and concept weight updating algorithms

user whether he/she had found the right answer or not. If the right answer is found, we recorded the AP for the query as 1, otherwise 0. We averaged all the APs as Mean Average Precision (MAP) to record the overall performance.

In the first experiment, we illustrate the effectiveness of the visual query and concept weight updating algorithms. We performed the interactive search on two settings: (1) we do not update the visual queries; and (2) we do not adjust the concept weights. The results are shown in Figure 7.8. First, the rectangle line indicates that the interactive video search is effective for MRVS tasks, where the MAPs are 0.958 and 1 respectively on "TV10" and "YT12" datasets after 10 minutes of interactive search, which means our system is able to find the answers for all of the queries except one in less than 10 minutes. We checked the failed query and found that the failure was due to incomplete text description associated with the right video and the incorrect visual and concept queries. Second, when visual queries are not updated, the search performance drops significantly, by 13% and 10% on "TV10" and "YT12" datasets respectively. This result shows that the visual query updating is able to correctly modify the features of visual queries to achieve a better performance. Third, when the system does not adjust concept weights, the search performance drops by 8.6% and 6% on "TV10" and "YT12" datasets respectively, which indicates that the concept weight updating algorithm

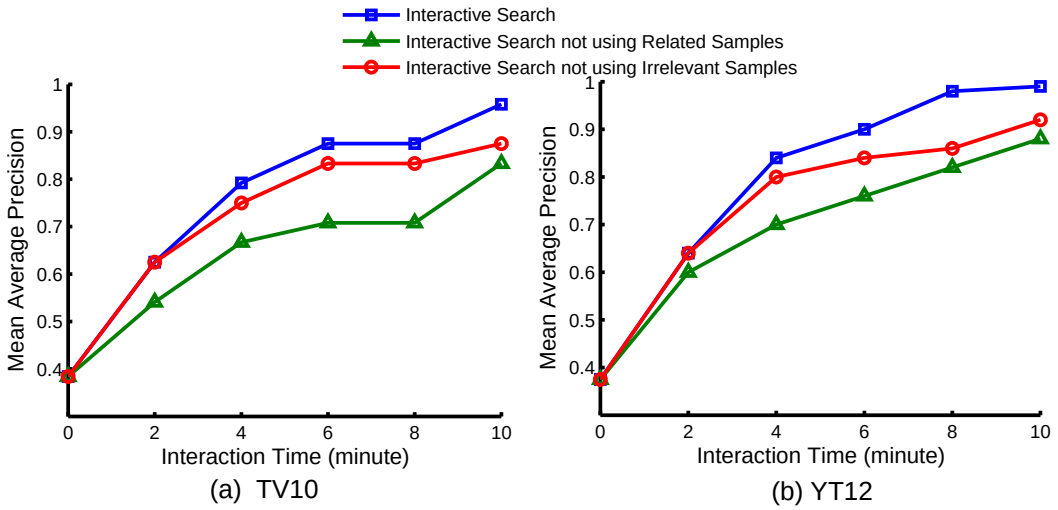


Figure 7.9: The comparison of video search performance by using or not using related samples.

is able to adjust the initial concept weights well to accelerate the performance improvement.

In the second experiment, we conducted the experiment to validate the effectiveness of using related and irrelevant samples. In this experiment, users were asked to perform the interactive search under three labeling schemes: (1) label both related and irrelevant samples, (2) label only irrelevant samples, and (3) label only related samples. The comparison results are shown in Figure 7.9. In Scheme (1), the MAPs are 0.958 and 1 on "TV10" and "YT12" datasets respectively after 10 minutes of interactive search. The corresponding values are 0.833, 0.88 respectively for Scheme (2), and 0.875, 0.92 respectively for Scheme (3). This result demonstrates that both related and irrelevant samples are useful to find the desired videos for users in interactive video search. The main reason is that the related samples are visually similar or semantically close to the relevant ones, while the irrelevant samples can be used to exclude irrelevant visual features and semantic concepts.

7.6 Conclusion

Memory recall based video search simulates a real world video search situation where a user wishes to find a desired video or video segments that he/she has seen before. In this paper, we developed a video search system that integrates text-based, content-based and semantic video search approaches for MRVS tasks. Given the approximate/incomplete recalls of the desired videos, we developed several innovative approaches to boost the performance of automatic video search. In particular, we developed a visual query suggestion module to help users refresh memory by suggesting better visual queries, proposed the use of a color similarity matrix to allow for inexact color matching, and proposed a reranking algorithm to exploit the temporal orders between visual and concept queries. To cater to the fact that there is often one or few relevant results, the system encourages users to perform relevance feedback by labeling related and irrelevant samples. In relevance feedback, we developed optimization algorithms to update the visual queries and concepts weights to refine the search results. We conducted experiments on two video datasets: TRECVID 2010 and YouTube 2012. The experimental results demonstrated the effectiveness of our system for MRVS tasks. In the future, we will incorporate situational context and other features such as motion vectors recalled by users to further enhance the search performance.

Chapter 8

Conclusions

In this chapter, we summarize our achievements in complex query learning. Also, a few potential research areas will be presented.

8.1 Summary of Research

This dissertation aims to enhance search performance for complex queries in semantic video search. We developed comprehensive methods for concept detection, automatic semantic video search, and interactive semantic video search. In addition, we applied and extended the proposed approaches in a real-world video search task named "Memory Recall based Video Search". Below, we summarize the contributions and findings of our work.

8.1.1 Concept Bundle Learning

In semantic video search, the simple aggregation of primitive concepts is unable to capture the relationship between the primitive concepts in a complex query. Therefore, in chapter 4, we moved a step ahead by proposing a high-level semantic descriptor named "Concept Bundle". Concept bundle is a composite concept that integrates multiple primitive concepts as well as the relationships between the concepts, such as ("lion", "hunting", "zebra"), ("police", "fighting", "protestor") etc.. Compared to the simple aggregation of primitive concepts, concept bundle is semantically closer to the meaning of complex query since it

contains both primitive concepts and the relationships between them. To build the classifiers of concept bundles, we first proposed an approach to automatically select informative concept bundle by measuring its frequency on the suggested queries by Web video search engine and the concept co-occurrence in the tags of Web videos. We then developed a multi-task SVM algorithm to effectively learn the classifiers for concept bundles. In particular, our multi-task SVM algorithm learns a concept bundle classifier based on both the training samples from its constituent primitive concepts and that from the concept bundle. The training samples of the constituent primitive concepts are used to model the individual concepts that appearing in the concept bundle, while that of the concept bundle is used to model the relationship between the concepts appearing in the concept bundle. We conducted experiments on "TV08" and "YT10" datasets. The results demonstrated that the proposed approaches could effectively select informative concept bundles, and achieve better performance to learn concept bundle classifiers as compared to the state-of-art methods.

8.1.2 Bundle-based Automatic Semantic Video Search

Given the primitive concepts and concept bundles, in chapter 5, we developed an optimization algorithm to map a query to related primitive concepts and concept bundles. To effectively perform concept selection, our algorithm considers two criteria: 1) semantic relatedness between the selected concepts and query; and 2) classifier performance of the selected concepts. Given a complex query, the concept selection algorithm prefers related concept bundles as compared to related primitive concepts incase when their respective classifiers have similar performance. This is because related concept bundles are semantically closer to the complex query. On the other hand, when the classifier of a primitive concept or concept bundle is poor, the algorithm would discard it to avoid noisy results. Based on these two criteria, we employed a greedy algorithm to approximately implement the optimization selection with the aim of saving the computational cost. The experiments were conducted on "TV08" and "YT10" datasets. The results showed that these two criteria can affect the search performance. In addition, as compared to the state-of-art approaches, the proposed concept selection

strategy achieved promising search performance.

8.1.3 Related Sample based Interactive Semantic Video Search

To further enhance the search performance, we incorporated interactive video search for complex queries. One challenging problem for complex queries in interactive video search is the sparse relevant sample problem, where a user may not be able to find sufficient relevant samples to label during the interactive process. Without sufficient relevant samples, the search performance is usually limited. To tackle this problem, we proposed a new sample class named "Related Sample". Related samples refer to those video segments that are partially relevant to the query but do not satisfy the entire search criterion. Compared to relevant samples which may be rare, related samples are usually more plentiful and easier to find in the search results. The advantages of exploring related samples are twofold: First, the related and relevant video segments usually share similar visual content in part due to their semantic connection, so that the related samples are beneficial to the modeling of relevant samples. Second, since video content is temporally dynamic and continuous, the occurrence of related video segments is an indicator for the presence of relevant ones in the neighboring clips. By exploring the visual and temporal attributes based on the labeled samples, we developed a visual-based ranking model and a temporal-based ranking model. Moreover, an adaptive fusion approach was used to learn the optimal fusion weight to generate the search results. We conducted experiments on "TV08" and "YT11" datasets. The experimental results demonstrated the related samples are effective to enhance search performance for complex queries in interactive video search.

8.1.4 Application: Memory Recall based Video Search

In Chapter 7, we applied and extended the proposed approaches in a novel video search task named "Memory Recall based Video Search" (MRVS). Memory recall based video search simulates a real-world video search situation that a user

wishes to find a desired video or video segments that he/she has been seen before. In our system, a user can input a text query, a sequence of visual and concept queries, or a combination of all based on his/her memory recall. Here, the text query is used to describe the textual information of the desired video; While the sequence of visual and concept queries is employed to depict the visual scenes in the desired video segments, where we organize them according to the temporal order. Based on the queries, the system returns the search results by integrating the results from text-based, content-based and semantic video search approaches. Specifically, since the visual queries are usually inaccurate based on a user's vague memory, we proposed a visual query suggestion approach to automatically suggest better visual queries, as well as a color similarity matrix to measure the color similarity between different colors. In semantic video search, we employed the proposed approaches on concept bundles to accurately interpret a user's query. Moreover, we proposed a sequence-based reranking algorithm to refine the search results from the content-based and semantic video search approaches by exploring the temporal order between the visual and concept queries. We further utilize the related sample approach in the interactive video search framework to overcome the extremely sparse relevant sample problem for MRVS task. By utilizing the visual similarity and semantical similarity between related and relevant samples, we proposed a visual query updating algorithm to modify the rough visual queries, as well as a concept weight updating algorithm to adjust the concept weights in semantic video search. These updated visual queries and concept weights are used to refine the search results by the automatic video search approach. We simulated the real-world video search situation based on users' memory recall in the experiments. The experiments were conducted on two video datasets: "TV10" and "YT12" datasets. The experimental results demonstrated our system is effective for MRVS.

8.2 Future Work

There are several limitations and potential extensions in the areas of research presented in this thesis.

- First, we developed a multi-task SVM algorithm to learn the classifier of a concept bundle based on the training samples from its constituent primitive concepts and the concept bundle. This approach assumes that all the training samples from the primitive concepts can help to model the semantic contributions of the primitive concepts in the concept bundle. However, given a relevant sample of a certain primitive concept, only a part of regions of the sample are useful to model the target concept bundle. Thus, effectively identifying related regions in the training samples may be useful to further enhance the classifier performance for concept bundle.

- Second, since the number of the pre-built concept bundles is limited, it does not ensure that all the issued complex queries in the real case are able to be mapped to the related concept bundles by the bundle-based semantic video search. In such case, the search performance may be unsatisfactory. Therefore, how to expand the concept bundle set to meet the demands of complex query search in the real world is an important direction to explore.

- Third, we proposed "Related Sample" to overcome the sparse relevant sample problem for complex queries in the interactive video search. By utilizing the visual similarity between related and relevant samples, we proposed a visual-based ranking model. However, given a related sample, only parts of the sample may be visually similar to the relevant samples. Therefore, extracting useful regions from the related samples may be more effective to finding relevant samples.

- Fourth, the related and relevant samples are visually dissimilar sometimes. For example, a user selects the related samples which satisfy the condition "*one or more colored photographs*", and the query is "*one or more black and white photographs*". In such case, the visual features of related and relevant samples are completely different. Thus, the use of the visual-based ranking model may degrade the search performance. In future, it is better to develop an approach to automatically identify the effectiveness of visual features in related samples.

8.3 Publications

We list the publications for this research as follows:

1. **Jin Yuan**, Zheng-Jun Zha, Zheng Dong Zhao, Xiang Dong Zhou and Tat-Seng Chua, "*Utilizing Related Samples to Learn Complex Queries in Interactive Concept-based Video Search*", Proc. of ACM Int. Conf. on Image and Video Retrieval, full paper (Oral), 2010.

2. **Jin Yuan**, Zheng-Jun Zha, Yan-Tao Zheng, Meng Wang, Xiang Dong Zhou and Tat-Seng Chua, "*Learning Concept Bundles for Video Search with Complex Queries*", Proc. of ACM Int. Conf. on Multimedia, full paper(Oral), 2011.

3. **Jin Yuan**, Zheng-Jun Zha, Yan-Tao Zheng, Meng Wang, Xiang Dong Zhou, and Tat-Seng Chua, "*Utilizing Related Samples to Enhance Interactive Concept-Based Video Search*", IEEE Transactions on Multimedia, volume 13, page 1343 - 1355, 2011.

4. **Jin Yuan**, Huanbo Luan, Dejun Hou, Han Zhang, Yan-Tao Zheng, Zheng-Jun Zha, and Tat-Seng Chua, "*Video Browser ShowDown by NUS*", Proc. of ACM Int. Conf. on Multimedia Modeling, 2012.

References

[AACea05] A. Amir, J. Argillandery, M. Campbellz, and et al. Ibm research trecvid-2005 video retrieval system. *In TRECVID Workshop*, 2005. 38

[ABC+03] A. Amir, M. Berg, S.-F. Chang, W. Hsu, and et al. Ibm research trecvid-2003 video retrieval system. *in Proceedings of the TRECVID Workshop*, 2003. 16

[AHO07] R. Aly, D. Hiemstra, and R. Ordelman. Building detectors to support searches on combined semantic concepts. *Proc. of the SIGIR workshop on Multimedia Information Retrieval*, 2007. 57, 60

[AZ05] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning*, 6:1817–1853, 2005. 26, 62

[Bis06] C. M. Bishop. Pattern recognition and machine learning. *Information Science and Statistics, Springer*, 2006. 6

[BLJ04] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. *Proc. of Int. Conf. on Machine Learning*, 2004. 58, 84, 115

[BMM99] R. Brunelli, O. Mich, and C. M. Modena. A survey on the automatic indexing of video data. *Journal of Visual Communication and Image Representation*, 10:78–112, 1999. 5

[Bri03] K. Brinker. Incorporating diversity in active learning with support vector machines. *Proc. of ACM Int. Conf. on Machine Learning*, 2003. 33

[BUB12] D. Borth, A. Ulges, and T. M. Breuel. Dynamic vocabularies for web-based concept detection by trend discovery. *In Proc. of the ACM Int. Conf. on Multimedia*, 2012. 22

[CB98] Christopher and J. C. Burges. A tutorial on support vector machines for pattern recognition. *DATA MINING AND KNOWLEDGE DISCOVERY*, 2:121–167, 1998. 6, 17, 18, 38

[CCHW05] M.-Y. Chen, M. G. Christel, A. G. Hauptmann, and H. Wactlar. Putting active learning into multimedia applications: Dynamic definition and refinement of concept classifiers. *Proc. of ACM Int. Conf. on Multimedia*, 2005. 33, 36

[CH07] M. G. Christel and A. G. Hauptmann. Exploring concept selection strategies for interactive video search. *Proc. of ACM Int. Conf. on Semantic Computing*, 2007. 33

[CHEea06] M. Campbell, A. Hauboldy, S. Ebadollahi, and et al. Ibm research trecvid-2006 video retrieval system. *TRECVID 2006 Workshop*, 2006. 38

[CHJ+06] S.-F. Chang, W. Hsu, W. Jiang, L. S. Kennedy, D. Xu, A. Yanagawa, and E. Zavesky. Columbia university trecvid-2006 video search and high-level feature extraction. *TRECVID Workshop*, 2006. 4, 6, 7, 12, 27, 30, 32, 38, 39, 68, 98

[CNZea06] T.-S. Chua, S.-Y. Neo, Y.-T. Zheng, and et al. Trecvid 2006 by nus-i2r. *TRECVID Workshop*, 2006. 38

[CTH+09] T.-S. Chua, J.-H Tang, R. C. Hong, H. J. Li, Z. P. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of singapore. *Proc. of ACM Int. Conf. on Image and Video Retrieval*, 2009. 44

[CWZea10] L. Chaisorn, K.-W Wan, Y.-T. Zheng, and et. al. Trecvid 2010 known-item search (kis) task by i2r. *In TRECVID Workshop*, 2010. 102

[CYNea10] X. Y. Chen, J. Yuan, L. Q. Nie, and et. al. Trecvid 2010 known-item search by nus. *In TRECVID Workshop*, 2010. 102

[CZC06] O. Chapelle, A. Zien, and B. Cholkopf. Semi-supervised learning. *MIT Press*, 2006. 23

[dRSW07] O. de Rooij, C. G. M. Snoek, and M. Worring. Query on demand video browsing. *Proc. of ACM Int. Conf. on Multimedia*, 2007. 37

[dRSW08] O. de Rooij, C. G. M. Snoek, and M. Worring. Balancing thread based navigation for targeted video search. *Proc. of ACM Int. Conf. on Image and Video Retrieval*, 2008. xi, 35, 37

[DSP91] G. Davenport, T. G. A. Smith, and N. Pincever. Cinematic principles for multimedia. *IEEE Computer Graphics & Applications*, 11:67–74, 1991. 17

[EP04] T. Evgeniou and M. Pontil. Regularized multi-task learning. *Proc. of the ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, 2004. 26

[Fai05] M. D. Fairchild. Color appearance models. *2nd edition. Addison-Wesley*, 2005. 107, 108

[Fel98] C. Fellbaum. Wordnet: an electronic lexical database. *The MIT Press*, 1998. 28, 60

[GBSG01] J.-M. Geusebroek, R. Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1338–1350, 2001. 17

[GCL04] K. S. Goh, Edward Y. Chang, and W. C. Lai. Multimodal concept dependent active learning for image retrieval. *Proc. of ACM Int. Conf. on Multimedia*, 2004. 33, 36

[GKS00] U. Gargi, R. Kasturi, and S. H. Strayer. Performance characterization of video-shot-change detection methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 10:1–13, 2000. 5, 17, 79

[GLT+12] B. Geng, Y. X. Li, D. C. Tao, M. Wang, Z.-J Zha, and C. Xu. Parallel lasso for large-scale video concept detection. *IEEE Transactions on Multimedia*, 14:55–65, 2012. 22

[GN08] P. Geetha and V. Narayanan. A survey of content-based video retrieval. *Journal of Computer Science*, 4:474–486, 2008. 38

[GS99] T. Gevers and A. W. M. Smeulders. Color-based object recognition. *Pattern Recognition*, 32:453–464, 1999. 17

[HAH07] C. Hauff, R. Aly, and D. Hiemstra. The effectiveness of concept based search for video retrieval. *In Workshop Information Retrieval*, 2007. 28, 31, 32

[Han02] A. Hanjalic. Shot-boundary detection: unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12:90–105, 2002. 5

[HBCea03] A. G. Hauptmann, R. V. Baron, M.-Y. Chen, and et al. Informedia at trecvid-2003: Analyzing and searching broadcast news video. *TRECVID Workshop*, 2003. 20

[Hes69] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, pages 303–320, 1969. 114

[HJL06] S. C. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. *Proc. of IEEE Int. Conf. on World Wide Web conference*, 2006. 33

[HL05] A. Hauptmann and W.-H. Lin. Assessing effectiveness in video retrieval. *Proc. of the ACM Int. Conf. on Image and Video Retrieval*, 2005. 73

[HLRYC06] A. G. Hauptmann, W.-H. Lin, J. Yang R. Yan, and M.-Y. Chen. Extreme video retrieval: joint maximization of human and computer performance. *Proc. of ACM Int. Conf. on Multimedia*, 2006. 8, 34, 36, 98

[HNN06] A. Haubold, A. Natsev, and M. R. Naphade. Semantic multimedia retrieval using lexical query expansion and model-based reranking. *Proc. of ACM Int. Conf. on Multimedia and Expo*, 2006. 28, 31, 32

[HXLZ11] W. M. Hu, N. H. Xie, L. Li, and X. L. Zeng. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on system, Man and Cybernetics, Part C: Applications and Reviews*, 41:797–819, 2011. 38

[HYea07] A. Hauptmann, Y. R. Yan, and et al. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *IEEE Transactions on Multimedia*, 9:958–966, 2007. 2, 38

[JCJL08] W. Jiang, S.-F. Chang, T. Jebara, and A. C. Loui. Semantic concept classification by joint semi-supervised learning of feature subspaces and support vector machines. *IEEE European Conf. on Computer Vision*, 2008. 17

[JDM00] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000. 17

[JF91] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24:1167–1186, 1991. 17

[JNC09] Y.-G. Jiang, C.-W. Ngo, and S.-F. Chang. Semantic context transfer across heterogeneous sources for domain adaptive video search. *Proc. of ACM Int. Conf. on Multimedia*, 2009. 2, 29, 32, 74, 75

[JWCN09] Y.-G Jiang, J. Wang, S.-F. Chang, and C.-W. Ngo. Domain adaptive semantic diffusion for large scale context-based video annota-

tion. *Proc. of ACM Int. Conf. on Computer Vision*, 2009. 25, 26

[JYNH10] Y.-G. Jiang, J. Yang, C.-W. Ngo, and A. G. Hauptmann. Representations of keypoint-based semantic concept detection: A comprehensive study. *IEEE Transactions on Multimedia*, 12:42–53, 2010. 4, 16, 17, 26

[KHDM98] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998. 19, 21

[KNC05] L. S. Kennedy, A. P. Natsev, and S.-F. Chang. Automatic discovery of query-class-dependent models for multimodal search. *Proc. of ACM Int. Conf. on Multimedia*, 2005. 38, 39

[KR08] M. Kankanhalli and Y. Rui. Application potential of multimedia information retrieval. *Proceedings of the IEEE*, 96:712–720, 2008. 2

[LH02] W.-H. Lin and A. G. Hauptmann. News video classification using svm-based multimodal classifiers and combination strategies. *Proc. of ACM Int. Conf. on Multimedia*, 2002. 19, 20

[LLE00] L. J. Latecki, R. Lakaemper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 424–429, 2000. 17

[Lu01] G. Lu. Indexing and retrieval of audio: A survey. *Multimedia Tools and Applications*, 15:269–290, 2001. 17

[Luc] Lucene. Lucene. *http://lucene.apache.org/java/docs/index.html*. 80

[LWLZ07] X. Li, D. Wang, J. Li, and B. Zhang. Video search in concept subspace: A text-like paradigm. *Proc. of Int. Conf. on Image and Video Retrieval*, 2007. 29, 32

[LZN+08] H. B. Luan, Y.-T. Zheng, S.-Y. Neo, Y. D. Zhang, S. X. Lin, and T.-S. Chua. Adaptive multiple feedback strategies for interactive video search. *Proc. of ACM Int. Conf. on Image and Video Retrieval*, 2008. xi, 2, 33, 36, 37

[MM96] B. S. Manjunath and W.-Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:836–842, 1996. 17

[MMM98] E. Moxley, T. Mei, and B. S. Manjunath. Video annotation through search and graph reinforcement mining. *IEEE Transactions on Multimedia*, 12:184–193, 1998. 25

[MRS09] C. D. Manning, P. Raghavan, and H. Schtze. *An Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England, 2009. 17, 55, 107

[MS99] C. Manning and H. Schutze. Foundations of statistical natural language processing. *MIT Press*, 1999. 68

[MZLea08] T. Mei, Z. J. Zha, Y. Liu, and et al. Msra att trecvid 2008: High-level feature extraction and automatic search. *In TRECVID Workshop*, 2008. 12

[NA01] R. Nicholas and M. Andrew. Toward optimal active learning through sampling estimation of error reduction. *Proc. of ACM Int. Conf. on Machine Learning*, 2001. 33

[NH01] M. R. Naphade and T. S. Huang. A probabilistic framework for semantic video indexing, filtering and retrieval. *IEEE Transactions on Multimedia*, 3:141–151, 2001. 16

[NHT+07] A. P. Natsev, A. Haubold, J. Tesic, L. Xie, and R. Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. *Proc. of Int. Conf. on Multimedia*, pages 991–1000, 2007. 7, 27, 30, 32

[NKH02] M. R. Naphade, I. V. Kozintsev, and T. S. Huang. Factor graph framework for semantic video indexing. *IEEE Transactions on Circuits and Systems for Video Technology*, 12, 2002. 12

[NS06] M. Naphade and J. R. Smith. Large-scale concept ontology for multimedia. *IEEE Transactions on Multimedia*, 13:86–91, 2006. 4, 17

[NWZ+11] L. Q. Nie, M. Wang, Z.-J. Zha, G. D. Li, and T.-S. Chua. Multimedia answering: Enriching text qa with media information. *Proc. of Int. Conf. on SIGIR*, pages 695–704, 2011. 102

[NZKC06] S.-Y. Neo, J. Zhao, M.-Y. Kan, and T.-S. Chua. Video retrieval using high level features: Exploiting query matching and confidence-based weighting. *Proc. of ACM Int. Conf. on Image and Video Retrieval*, 2006. 12, 28, 31, 32

[PACG08] J. Pickens, J. Adcock, M. Cooper, and A. Girgensohn. Fxpal interactive search experiments for trecvid 2008. *TRECvid Working Notes*, 2008. 4, 34

[Pla00] J. Platt. Advances in large margin classifiers, chapter probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *MIT Press*, 2000. 18

[QHR+07] G.-J. Qi, X.-S. Hua, Y. Rui, J. H. Tang, T. Mei, and H.-J. Zhang. Correlative multi-label video annotation. *Proc. of ACM Int. Conf. on Multimedia*, 2007. 2, 12, 22, 26, 62

[RHOM98] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8:644–655, 1998. 33

[ROS04] M. Rautiainen, T. Ojala, and T. Seppanen. Cluster-temporal browsing of large news video databases. *Proc. of ACM Int. Conf. on Multimedia and Expo*, 2004. xi, 35, 37

[SBB+12] S. T. Strat, A. Benoit, H. Bredin, G. Quenot, and P. Lambert. Hierarchical late fusion for concept detection in videos. *ECCV Workshop on Information Fusion in Computer Vision for Concept Recognition*, 2012. 21

[SC96] J. R. Smith and S.-F. Chang. Searching for images and videos on the world-wide web. *IEEE Multimedia Magazine*, 1996. 2, 37

[SC97] J. R. Smith and S.-F. Chang. Visually searching the web for content. *IEEE MultiMedia*, 4:12–20, 1997. 16

[SEZ05] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: Video shot retrieval for dace sets. *In Proc. Int. Conf. Image Video Retrieval*, 2005. 38

[SHHe07] C. G. M. Snoek, B. Huurnink, L. Hollink, and et.al. Adding semantics to detectors for video retrieval. *IEEE Transactions on Multimedia*, 9, 2007. 28, 29, 31, 32

[SN03] J. R. Smith and M. Naphade. Multimedia semantic indexing using model vectors. *Proc. of the IEEE Int. Conf. on Multimedia and Expo*, 2003. 2, 22, 26, 29, 32

[SP98] M. Szummer and R. W. Picard. Indoor-outdoor image classification. *IEEE International Workshop on Content-based Access of Image and Video Databases*, 1998. 16

[SvdSdR+08] C. G. M. Snoek, K. E. A. van de Sande, O. de Rooij, B. Huurnink, J. C. van Gemert, and et al. The mediamill trecvid 2008 semantic video search engine. *TRECvid Working Notes*, 2008. 32, 38

[SvGGea06] C. G. M. Snoek, J. C. van Gemert, T. Gevers, and et al. The mediamill trecvid 2006 semantic video search engine. *TRECVID Workshop*, 2006. 19, 20, 26

[SW09] C. G. M. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends ®in Information Retrieval*, 2:215–322, 2009. 2, 6, 7, 72, 77

[SWG+06a] C. G. M. Snoek, M. Worring, J. C. V. Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. *Proc. of ACM Int. Conf. on Multimedia*, 2006. 4, 6, 16, 17

[SWG+06b] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, F. J. Seinstra, and A. W. M. Smeulders. The semantic pathfinder: Using an authoring metaphor for generic multimedia indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1678–1689, 2006. 16

[SWH06] C. G. M. Snoek, M. Worring, and A. G. Hauptmann. Learning rich semantics from news video archives by style analysis. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2:91–108, 2006. 20

[SWKS07] C. G. M. Snoek, M. Worring, D. C. Koelma, and A. W. M. Smeulders. A learned lexicon-driven paradigm for interactive video retrieval. *IEEE Transactions on Multimedia*, pages 280–292, 2007. 37

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Proc. of ACM Magazine Communications*, 18:613–620, 1975. 28

[TC01] S. Tong and E. Chang. Support vector machine active learning for image retrieval. *Proc. of ACM Int. Conf. on Multimedia*, 2001. 33

[THL+05] H. Tong, J. R. He, M. J. Li, C. S. Zhang, and W. Y. Ma. Graph-based multi-modality learning. *Proc. of ACM Int. Conf. on Multimedia*, 2005. 24, 26

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society.*, 58:267–288, 1996. 22

[TLea03] B. L. Tseng, C. Y. Lin, and et al. Normalized classifier fusion for semantic visual concept detection. *Proc. of IEEE Int. Conf. on Image Processing*, pages 535–538, 2003. 19, 20, 26

[TLea08] S. Tang, J. T. Li, and et al. Trecvid 2008 high-level feature extraction by mcg-ict-cas. *TRECVID Workshop*, 2008. 17, 44

[TRE] TRECVID2010. Trecvid2010. *http://www-nlpir.nist.gov/projects/tv2010/tv2010.html*, page 2010. 47, 102

[TRE08] TRECVID. http://trecvid.nist.gov/. 2008. 40, 62, 72, 74

[TRSR09] P. Toharia, O. D. Robles, A. F. Smeaton, and A. Rodriguez. Measuring the influence of concept detection on video retrieval. *Proc. of ACM Int. Conf. on Computer Analysis of Images and Patterns*, 2009. 8, 34, 36

[TTR12] X. M. Tian, X. M. Tian, and Y. Rui. Sparse transfer learning for interactive video search reranking. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 8:520–540, 2012. 32

[VH01] R. C. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. *in Principles of Visual Information Retrieval*, pages 87–119, 2001. 17

[WC08] M.-F. Weng and Y.-Y. Chuang. Multi-cue fusion for semantic video indexing. *Proc. of ACM Int. Conf. on Multimedia*, 2008. 24, 26

[WC12] M.F Weng and Y.-Y Chuang. Cross-domain multicue fusion for concept-based video indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1927–1941, 2012. 25

[WCCS04] Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith. Optimal multimodal fusion for multimedia data analysis. *Proc. of ACM Int. Conf. on Multimedia*, pages 572–579, 2004. 20

[WHHea09] M. Wang, X.-S. Hua, R.-C. Hong, and et al. Unified video annotation via multi-graph learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 19:733–746, 2009. 17, 24, 26

[WHSea06] M. Wang, X.-S. Hua, Y. Song, and et al. Automatic video annotation by semi-supervised learning with kernel density estimation. *Proc. of ACM Int. Conf. on Multimedia*, 2006. 24, 26

[WLLZ07] D. Wang, X. Li, J. Li, and B. Zhang. The importance of query concept mapping for automatic video retrieval. *Proc. of Int. Conf. on Multimedia*, pages 285–288, 2007. 6, 7, 31

[WMC09] K. Wang, Z. Y. Ming, and T.-S. Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. *Proc. of the 32nd ACM SIGIR*, 2009. 68

[WN08] X.-Y. Wei and C.-W. Ngo. Fusing semantics, observability, reliability and diversity of concept detectors for video search. *Proc. of the ACM Int. Conf. on Multimedia*, 2008. 74, 75

[WNJ08] X.-Y. Wei, C.-W. Ngo, and Y.-G. Jiang. Selection of concept detectors for video search by ontology-enriched semantic spaces. *IEEE Transactions on Multimedia*, 10:1085–1096, 2008. 4

[WP94] Z. Wu and M. Palmer. Verbs semantics and lexical selection. *Proc. of ACM Int. Conf. on Association for Computational Linguistics*, 1994. 28

[WTS04] Y. Wu, B. L. Tseng, and J. R. Smith. Ontology-based multi-classification learning for video concept detection. *IEEE Int. Con.on Multimedia and Expo*, 2004. 12, 22, 26

[WWL+08] D. Wang, Z. K. Wang, J. M. Li, B. Zhang, and X. R. Li. Query representation by structured concept threads with application to interactive video retrieval. *Journal of Visual Communication and Image Representation*, 20:104–116, 2008. 31, 32

[WWLZ08] Z. K. Wang, D. Wang, J. M. Li, and B. Zhang. Learning structured concept-segments for interactive video retrieval. *Proc. of ACM Int. Conf. on Image and Video Retrieval*, 2008. 2, 34, 36

[WZP00] Y. Wu, Y. Zhuang, and Y. Pan. Content-based video similarity model. *In Proc. of the ACM Int. Conf. on Multimedia*, 2000. 38

[YCKH07] A. Yanagawa, S.-F. Chang, L. Kennedy, and W. Hsu. Columbia university's baseline detectors for 374 lscom semantic visual concepts. *ADVENT Technical Report 222-2006-8*, 2007. 4, 6, 12, 16, 19, 20, 26, 40, 41, 91

[YHC04] H. Yu, J. W. Han, and K. C.-C Chang. Pebl: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16:70–81, 2004. 38

[YHJ03] R. Yan, A. Hauptmann, and R. Jin. Multimedia search with pseudo-relevance feedback. *In Proc. of the ACM Int. Conf. on Image and Video Retrieval*, pages 238–247, 2003. 25

[You12] YouTube. Youtube statistic. *http://www.intel.com/content/www/us/en/communications/internet-minute-infographic.html*, 2012. 1

[YY10] X. T. Yuan and S. C. Yan. Visual classification with multi-task joint sparse representation. *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2010. 26

[YZZ+10] J. Yuan, Z.-J. Zha, Z. D. Zhao, X. D. Zhou, and T.-S. Chua. Utilizing related samples to learn complex queries in interactive concept-based video search. *Proc. of ACM Int. Conf. on Image and Video Retrieval*, 2010. 92, 98, 99

[YZZ+11a] J. Yuan, Z.-J. Zha, Y.-T. Zheng, M. Wang, X. D. Zhou, and T.-S. Chua. Learning concept bundles for video search with complex queries. *Proc. of ACM Int. Conf. on Multimedia*, 2011. 103, 109, 114, 115

[YZZ+11b] J. Yuan, Z.-J. Zha, Y.-T. Zheng, M. Wang, X. D. Zhou, and T.-S. Chua. Utilizing related samples to enhance interactive concept-based video search. *IEEE Transactions on Multimedia*, 13:1343–1355, 2011. 104, 112

[Zhu05] X. J. Zhu. Semi-supervised learning with graphs. *Doctoral thesis, CMU*, 2005. 17, 23, 89

[ZNCC09] Y.-T. Zheng, S.-Y. Neo, X. Y. Chen, and T.-S. Chua. Visiongo: towards true interactivity. *Proc. of ACM Int. Conf. on Image and Video Retrieal*, 2009. 4

[ZPYP08] V. W. Zheng, S. J. Pan, Q. Yang, and J. J. Pan. Transferring multi-device localization models using latent multi-task learning. *Proc. of the Int. Conf. on Artificial intelligence*, 2008. 26

[ZTSG95] H.-J. Zhang, S. Y. Tan, S. W. Smoliar, and Y. Gong. Automatic parsing and indexing of news video. *Multimedia Systems*, 2:256–266, 1995. 16

[ZWZ+12] Z.-J. Zha, M. Wang, Y.-T. Zheng, Y. Yang, R.-C. Hong, and T.-S. Chua. Interactive video indexing with statistical active learning. *IEEE Transactions on Multimedia*, 14:17–27, 2012. 33