CARNEGIE MELLON UNIVERSITY

Web-scale Multimedia Search for Internet Video Content

Lu Jiang

CMU-LTI-17-003

Language and Information Technologies
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA, 15213
www.lti.cs.cmu.edu

Thesis Committee:
Dr. Alex Hauptmann, Carnegie Mellon University
Dr. Teruko Mitamura, Carnegie Mellon University
Dr. Louis-Philippe Morency, Carnegie Mellon University
Dr. Tat-Seng Chua, National University of Singapore

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies

“宝剑锋从磨砺出，梅花香自苦寒来。”

“April showers bring May flowers.”

《警世贤文》之勤奋篇

Cautionary words of wisdom

Abstract

The Internet has been witnessing an explosion of video content. According to a Cisco study, video content accounted for 64% of all the world's internet traffic in 2014, and this percentage is estimated to reach 80% by 2019. Video data are becoming one of the most valuable sources to assess information and knowledge. However, existing video search solutions are still based on text matching (*text-to-text* search), and could fail for the huge volumes of videos that have little relevant metadata or no metadata at all. The need for large-scale and intelligent video search, which bridges the gap between the user's information need and the video content, seems to be urgent.

In this thesis, we propose an accurate, efficient and scalable search method for video content. As opposed to text matching, the proposed method relies on automatic video content understanding, and allows for intelligent and flexible search paradigms over the video content (*text-to-video* and *text&video-to-video* search). It provides a new way to look at content-based video search from finding a simple concept like "puppy" to searching a complex incident like "a scene in urban area where people running away after an explosion". To achieve this ambitious goal, we propose several novel methods focusing on accuracy, efficiency and scalability in the novel search paradigm. First, we introduce a novel self-paced curriculum learning theory that allows for training more accurate semantic concepts. Second, we propose a novel and scalable approach to index semantic concepts that can significantly improve the search efficiency with minimum accuracy loss. Third, we design a novel video reranking algorithm that can boost accuracy for video retrieval. Finally, we apply the proposed video engine to tackle text-and-visual question answering problem called MemexQA.

The extensive experiments demonstrate that the proposed methods are able to surpass state-of-the-art accuracy on multiple datasets. In addition, our method can efficiently scale up the search to hundreds of millions videos, and only takes about 0.2 second to search a semantic query on a collection of 100 million videos, 1 second to process a hybrid query over 1 million videos. Based on the proposed methods, we implement E-Lamp Lite, the first of its kind large-scale semantic search engine for Internet videos. According to National Institute of Standards and Technology (NIST), it achieved the best accuracy in the TRECVID Multimedia Event Detection (MED) 2013, 2014 and 2015, the most representative task for content-based video search. To the best of our knowledge, E-Lamp Lite is the first content-based semantic search system that is capable of indexing and searching a collection of 100 million videos.

Acknowledgements

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

This work was partially supported by Yahoo InMind project. Besides, the author would like to thank the Flickr Computer Vision and Machine Learning group for their kind support.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. It used the Blacklight system at the Pittsburgh Supercomputing Center (PSC).

Contents

Dedicated to the ones who raised me up to more than I can be.

Chapter 1

Introduction

We are living in an era of big data: three hundred hours of video are uploaded to YouTube every minute; social media users are posting 12 millions videos on Twitter every day. According to a Cisco study, video content accounted for 64% of all the world's internet traffic in 2014, and this percentage is estimated to reach 80% by 2019. The explosion of video data is creating impacts on many aspects of society. The big video data is important not because there is a lot of it but because increasingly it is becoming a valuable source for insights and information, e.g. telling us about things happening in the world, giving clues about a person's preferences, pointing out places, people or events of interest, providing evidence about activities that have taken place [1].

An important approach of acquiring information and knowledge is through video retrieval. However, existing large-scale video retrieval methods are still based on text-to-text matching, in which the query words are matched against the textual metadata generated by the uploader [2]. The text-to-text search method, though simple, is of minimum functionality because it provides no understanding about the video content. As a result, the method proves to be futile in many scenarios, in which the metadata are either missing or less relevant to the visual video content. Studies show that 66% videos on a social media site called Twitter Vine are not associated with meaningful metadata (hashtags or mentions) [3], and about 80% personal videos do not have any user tags [4]. This suggests on an average day, 8 millions of Twitter videos may never be watched again just because there is no good way to find them. The phenomenon is more severe for the even larger amount of videos that are captured by mobile phones, surveillance cameras and wearable devices that end up not having any metadata at all. Comparable to the days in the late 1990s, when people usually got lost in the rising sea of web pages, now they are overwhelmed by the vast amounts of videos, but lack powerful tools to discover, not to mention to analyze, meaningful information in the video content.

In this thesis, we seek the answer to a fundamental research question: how to satisfy information needs about video content at a very large scale. We embody this fundamental question into a concrete problem called Content-Based Video Semantic Retrieval (CBVSR), a category of content-based video retrieval problem focusing on semantic understanding about the video content, rather than on textual metadata nor on low-level statistical matching of color, edges, or the interest points in the content. A distinguishing characteristic about the CBVSR method is the capability to search and analyze videos based on semantic features that can be automatically extracted from the video content. The semantic features are human interpretable multimodal tags about the image or video content such as people (who were involved), objects (what objects were seen), scenes (where did it take place), actions and activities (what happened), speech (what did they say), visible text (what characters were spotted). *Semantic concepts*, or *concepts* for short, represent the entities discovered in the content of images or videos, including people, objects, scenes, actions, activities, etc. In this thesis, we assume semantic features include the semantic concept, speech, and visible text.

The CBVSR method advances traditional video retrieval methods in many ways. It enables a more intelligent and flexible search paradigm that traditional metadata search would never achieve. A simple query in CBVSR may contain a single object about, say, “a puppy” or “a desk”, and a regular query may describe a complex activity or incident, e.g. “changing a vehicle tire”, “attempting bike tricks in the forest”, “a group of people protesting an education bill”, “a scene in urban area where people running away after an explosion”, and so forth. In this thesis, we consider the following two types of queries:

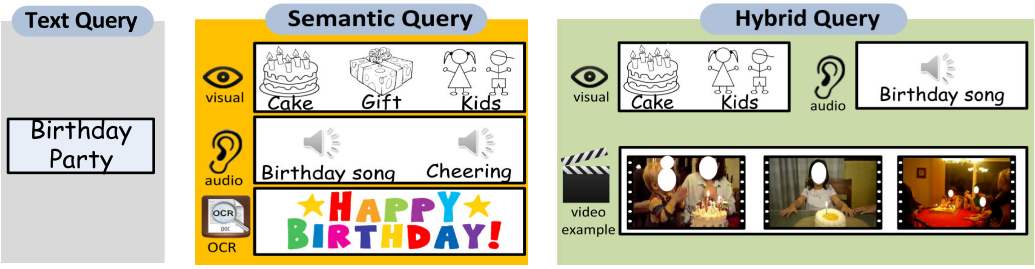
Definition 1.1. (Semantic Query and Hybrid Query) Queries only consisting of semantic features (e.g. people, objects, actions, speech, visible text, etc.) or a text description about semantic features are called *semantic queries*. Queries consisting of both semantic features and a few video examples are called *hybrid queries*. As video examples are usually provided by users on the fly, according to NIST [5], we assume there are at most 10 video examples in a hybrid query.

A user may formulate a semantic query in terms of a few semantic concept names or a natural language description of her information need (See Chapter 4). According to the definition, the semantic query provides an approach for text-to-video search, and the hybrid query offers a mean for text&video-to-video search. Semantic queries are important as, in a real-world scenario, users often start the search without any video example. A query consisting only of a few video examples is regarded as a special case of the hybrid query. Example 1.1 illustrates an example of formulating the queries for birthday party.

Example 1.1. *Suppose our goal is to search the videos about birthday party. In the traditional text query, we have to search the keywords in the user-generated metadata,* *such as titles and descriptions, as shown in Fig. 1.1(a). For videos without any metadata, there is no way to find them at all. In contrast, in a semantic query we might look for visual clues in the video content such as “cake”, “gift” and “kids”, audio clues like “birthday song” and “cheering sound”, or visible text like “happy birthday”. See Fig. 1.1(b). We may alternatively input a sentence like “videos about birthday party in which we can see cake, gift, and kids, and meanwhile hear birthday song and cheering sound.”*

Semantic queries are flexible and can be further refined by Boolean operators. For example, to capture only the outdoor party, we may add “AND outdoor’ to the current query; to exclude the birthday parties for a baby, we may add “AND NOT baby”. Temporal relation can also be specified by a temporal operator. For example, suppose we are only interested in the videos in which the opening of presents are seen before consuming the birthday cake. In this case, we can add a temporal operator to specify the temporal occurrence of the two objects “gift” and “cake”.

After watching some of the retrieved videos for a semantic query, the user is likely to select a few interesting videos, and to find more relevant videos like these [6]. This can be achieved by issuing a hybrid query which adds the selected videos to the query. See Fig. 1.1(c). Users may also change the semantic features in the hybrid query to refine or emphasize certain aspects in the selected video examples. For example, we may add “AND birthday song” in the hybrid query to find more videos not only similar to the video examples but also have happy birthday songs in their content.



(a) Text query (b) Semantic Query (c) Hybrid Query

FIGURE 1.1: Comparison of text, semantic and hybrid query on “birthday party”.

1.1 Research Challenges and Solutions

The idea of CBVSR sounds appealing but, in fact, it is a very challenging problem. It introduces several issues that have not been sufficiently studied in the literature, such as the issue of searching complex query consisting of multimodal semantic features and video examples, the novel search paradigm entirely based on video content understanding, and efficiency issue for web-scale video retrieval. As far as this thesis is concerned, we confront the following research challenges:

1. **Challenges on accurate retrieval for complex queries**. A crucial challenge for any retrieval system is achieving a reasonable accuracy, especially for the top-ranked documents or videos. Unlike other problems, the data in this problem are real-world noisy and complex Internet videos, and the queries are of complex structures containing both texts and video examples. How to design intelligent algorithms to obtain state-of-the-art accuracy is a challenging issue.
2. **Challenges on efficient retrieval at very large scale**. Processing video proves to be a computationally expensive operation. The huge volumes of Internet video data brings up a key research challenge. How to design efficient algorithms that are able to search hundreds of millions of video within the maximum recommended waiting time for a user, i.e. 2 seconds [7], while maintaining maximum accuracy becomes a critical challenge.
3. **Challenges on interpretable results**. A distinguishing characteristic about CBVSR is that the retrieval is entirely based on semantic understanding about the video content. A user should have some understanding of why the relevant videos are selected, so that she can modify the query to better satisfy her information need. In order to produce accountable results, the model should be interpretable. However, how to build interpretable models for content-based video retrieval is still unclear in the literature.

Due to the recent advances in the fields of computer vision, machine learning, and multimedia, it becomes increasingly interesting to consider addressing the above research challenges. In analogy to building a rocket spaceship, we are now equipped with powerful cloud computing infrastructures (structural frame) and big data (fuel). What is missing is a rocket engine that provides driving force and reaches the target. In our problem, the engine is essentially a collection of effective algorithms that can solve the above challenges. To this end, we propose the following novel methods:

1. To address the challenges on accuracy, we explore the following aspects. In Chapter 4, we systematically study a number of query generation methods, which translate a user query to a system query that can be handled by the system, and retrieval algorithms to improve the accuracy for semantic query. In Chapter 6, we propose a cost-effective reranking algorithm called self-paced reranking. It optimizes a concise mathematical objective and provides notable improvement for both semantic and hybrid queries. In Chapter 7, we propose a theory of self-paced curriculum learning, and then apply it to training more accurate semantic concept detectors.
2. To address the challenges on efficiency and scalability, in Chapter 3 we propose a semantic concept adjustment and indexing algorithm that provides a foundation for efficient search over 100 millions of videos. In Chapter 5, we propose a search algorithm for hybrid queries that can efficiently search a large-scale video collection, without significant loss on accuracy.

3. To address the challenges on interpretability, we design algorithms to build interpretable models based on semantic (and latent semantic) features. In Chapter 4, we provide a semantic justification that can explain the reasoning of selecting relevant videos for the semantic query. In Chapter 5, we discuss an approach that can explain the reasoning behind the search for results retrieved by a hybrid query.

The above proposed methods are extensively verified on a number of large-scale challenging datasets. Experimental results demonstrate that the proposed method can exceed state-of-the-art accuracy across a number of datasets. Furthermore, it can efficiently scale up the search to hundreds of millions of Internet videos. It only takes about 0.2 second to search a semantic query on a collection of 100 million videos, and 1 second to handle a hybrid query over 1 million videos.

Based on the proposed methods, we implement E-Lamp Lite, the first of its kind large-scale semantic search engine for Internet videos. According to National Institute of Standards and Technology (NIST), it achieved the best accuracy in the TRECVID Multimedia Event Detection (MED) 2013, 2014, and 2015, one of the most representative and challenging tasks for content-based video search. To the best of our knowledge, E-Lamp Lite is also the first content-based video retrieval system that is capable of indexing and searching a collection of 100 million videos.

1.2 Social Validity

The problem studied in this thesis is fundamental. The proposed methods can potentially benefit a variety of related tasks such as video summarization [8], video recommendation, video hyperlinking [9], social media video stream analysis [10], in-video advertising [11], etc. A direct usage is augmenting existing metadata search paradigms for video. Our method provides a solution to control video pollution on the web [12], which results from introduction into the environment of (i) redundant, (ii) incorrect, noisy, imprecise, or manipulated, or (iii) undesired or unsolicited videos or meta-information (i.e., the contaminants). Another application is about in-video advertising. Currently, it may be hard to place in-video advertisements as the user-generated metadata typically does not describe the video content, let alone concept occurrences in time. Our method provides a solution by formulating this information need as a semantic query and putting ads into the relevant videos [11]. For example, a sport shoe company may use the query "(running OR jumping) AND parkour AND urban scene" to find parkour videos in which the promotional shoe ads can be put.

Furthermore, our method provides a feasible solution of finding information in the videos without any metadata. Analyzing video content helps automatically understanding about what happened in the real life of a person, an organization or even a country. This functionality is crucial for a variety of applications. For example, finding videos in social streams that violate either legal or moral standards; analyzing videos captured by a wearable device, such as Google Glass, to assist the user's cognitive process on a complex task [13]; searching specific events captured by surveillance cameras or even devices that record other of types of signals.

Finally, the theory and insights in this thesis may inspire the development of more advanced methods. For example, the insight in our web-scale method may guide the design of the future search or learning systems for video big data [14]. The proposed reranking method can be also used to improve the accuracy of image retrieval [15]. The self-paced curriculums learning theory may inspire other machine learning methods.

1.3 Thesis Overview

In this thesis, we model a CBVSR problem as a retrieval problem, in which given a query that complies with Definition 1.1, we are interested in finding a ranked list of relevant videos based on the semantic understanding about the video content. To solve this problem, we incorporate a two-stage framework as illustrated in Fig. 1.2.

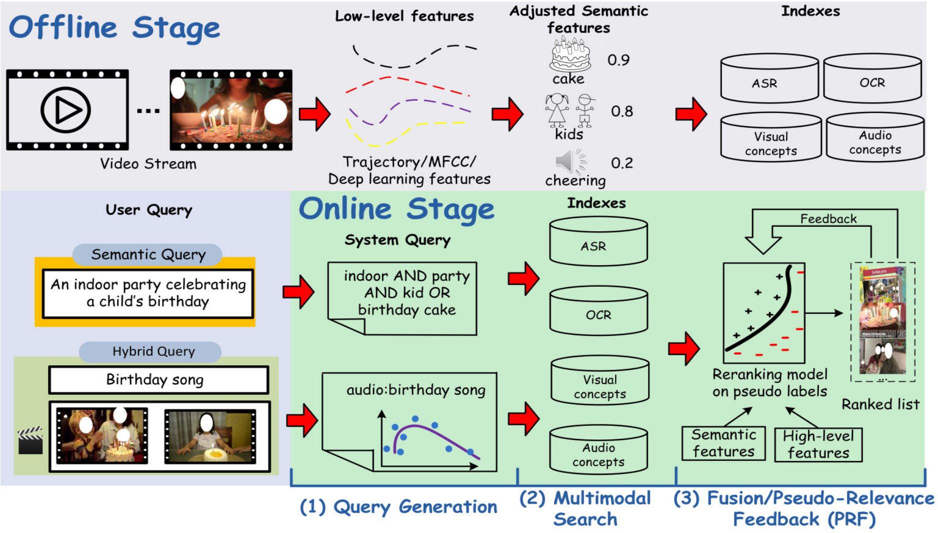


FIGURE 1.2: Overview of the framework for the proposed method.

The offline stage is called semantic indexing, which aims at extracting semantic features in the video content and indexing them for efficient online search. It usually involves the following steps: a video clip is first represented by the *low-level features* that capture the local appearance, texture or acoustic statistics in the video content, represented by a collection of local descriptors such as interest points or trajectories. State-of-the-art low-level features include dense trajectories [16] and convolutional Deep Neural

Network (DNN) features [17] for visual modality, and Mel-frequency cepstral coefficients (MFCCs) [18] and DNN features for audio modality [19, 20]. The low-level features are then input into the off-the-shelf detectors to extract the *semantic features*[1]. The semantic features, also known as high-level features, are human interpretable tags, each dimension of which corresponds to a confidence score of detecting a concept or a word in the video [21]. The visual/audio concepts, Automatic Speech Recognition (ASR) [19, 20, 22] and Optical Character Recognition (OCR) are four types of semantic features considered in this thesis. Semantic visual concepts, semantic audio concepts, ASR and OCR are four types of semantic features considered in this thesis. After extraction, the high-level features will be adjusted and indexed for the efficient online search. The offline stage can be trivially paralleled by distributing the videos over multiple cores[2].

The second stage is an online stage called video search. We employ two modules to process the semantic query and the hybrid query. Both modules consist of a query generation and a multimodal search step. A user can express a query in the form of a text description and a few video examples. The query generation for semantic query is to map the out-of-vocabulary concepts in the user query to their most relevant alternatives in the system vocabulary. For the hybrid query, the query generation also involves training a classification model using the video examples. The multimodal search component aims at retrieving a ranked list using the multimodal features. This step is a retrieval process for the semantic query and a classification process for the hybrid query. Afterwards, we can refine the results by reranking the videos in the initial ranked list. This process is known as reranking or Pseudo-Relevance Feedback (PRF) [25]. The basic idea is to first select a few videos and assign assumed labels to them. The samples with assumed labels are then used to build a reranking model using semantic and low-level features to improve the initial ranked list.

The quantity (relevance) and quality of the semantic concepts are two factors in affecting performance. The relevance is measured by the coverage of the concept vocabulary to the query, and thus is query-dependent. For convenience, we name it quantity as a larger vocabulary tends to increase the coverage. Quality determines the accuracy of the detector. To increase both the criteria, We propose a novel self-paced curriculum learning theory that allows for training more accurate semantic concepts over noisy datasets. The theory is inspired by the learning process of humans and animals that gradually proceeds from easy to more complex samples in training.

The reminder of this thesis will discuss the above topics in more details. In Chapter 2, we first briefly review related problems. In Chapter 3, we propose a scalable semantic

[1] Here we assume we are given the off-the-shelf detectors. Chapter 7 will introduce the approach to build the detectors.

[2] In this thesis, we do not discuss the offline video crawling process. This problem can be solved by the vertical crawling techniques [23, 24]

indexing and adjustment method for semantic feature indexing. We then discuss the multimodal search for semantic queries in Chapter 4, and the query embedding and search method for hybrid queries in Chapter 5. The multimodal reranking method will be discussed in 6. Finally we will introduce the method for training robust semantic concepts in Chapter 5. The conclusions and a demo MemexQA system will be presented in the last chapter.

1.4 Thesis Statement

In this thesis, we approach a fundamental problem of searching information in video content at a very large scale. We address the problem by proposing an accurate, efficient, and scalable method that can search the content of a billion videos by semantic concepts, speech, visible texts, video examples, or any combination of these elements.

1.5 Key Contributions of the Thesis

To summarize, the contributions of the thesis are as follows:

1. The first-of-its-kind framework for web-scale content-based search over hundreds of millions of Internet videos [ICMR'15, WWW'16]. The proposed framework supports text-to-video, video-to-video, and text&video-to-video search [MM'12].
2. A novel theory about self-paced curriculums learning and its application on robust concept detector training [NIPS'14, AAAI'15, IJCAI'16].
3. A novel reranking algorithm that is cost-effective in improving performance. It has a concise mathematical objective to optimize and useful properties that can be theoretically verified [MM'14, ICMR'14].
4. A consistent and scalable concept adjustment method representing a video by a few salient and consistent concepts that can be efficiently indexed by the modified inverted index [MM'15].
5. A novel query embedding for personal media search [WSDM'17], and a new joint learning model for the hybrid query.

Based on the above contributions, we implement E-Lamp Lite, the first of its kind large-scale semantic search engine for Internet videos. To the best of our knowledge, E-Lamp Lite is also the first content-based video retrieval system that is capable of indexing and searching a collection of 100 million videos. In Chapter 8, we demonstrate an interesting application that is built on the proposed E-Lamp Lite system. We show that the E-Lamp Lite can serves as an important fundamental engine for intelligent systems.

Chapter 2

Related Work

Traditional content-based video learning and retrieval methods have successfully been used to address a number of real-world problems. In this chapter, we briefly review some of these related problems. The goal of this chapter is to analyze their differences to the proposed problem.

2.1 Content-based Image Retrieval

Given a query image, a content-based image retrieval method is to find identical or visually similar images in a very large image collection. Similar images are "visually-alike" images about the same object despite possibly changes in scale, viewpoint, lighting and partial occlusion. It is a type of query-by-example search, where the query is usually represented by a single image, and can be extended to find a key frames in the video clip. As shown in Fig. 2.1, the query image is a coca-cola bottle and the results are coca-cola bottles captured at different angles. Content-based image retrieval is often used to search the images about a specific instance such as about a person, a logo or a landmark. In some cases, users can select a region of interest in an image, and use it as a query image [26].

Content-based image retrieval problem is a well-studied problem, and there have existed a number of commercial applications such as Google and Bing Image Search. State-of-the-art image retrieval systems can efficiently handle billions of images [27]. Generally, the solution is to first extract the low-level descriptors of an image such as SIFT [28], encode them into a numerical vector by, for example, bag-of-visual-words [29] or fisher vector [30], and finally index the feature vectors for efficient online search using min-hashing or LSH [31]. For example, Sivic et al. introduced a video frame retrieval system

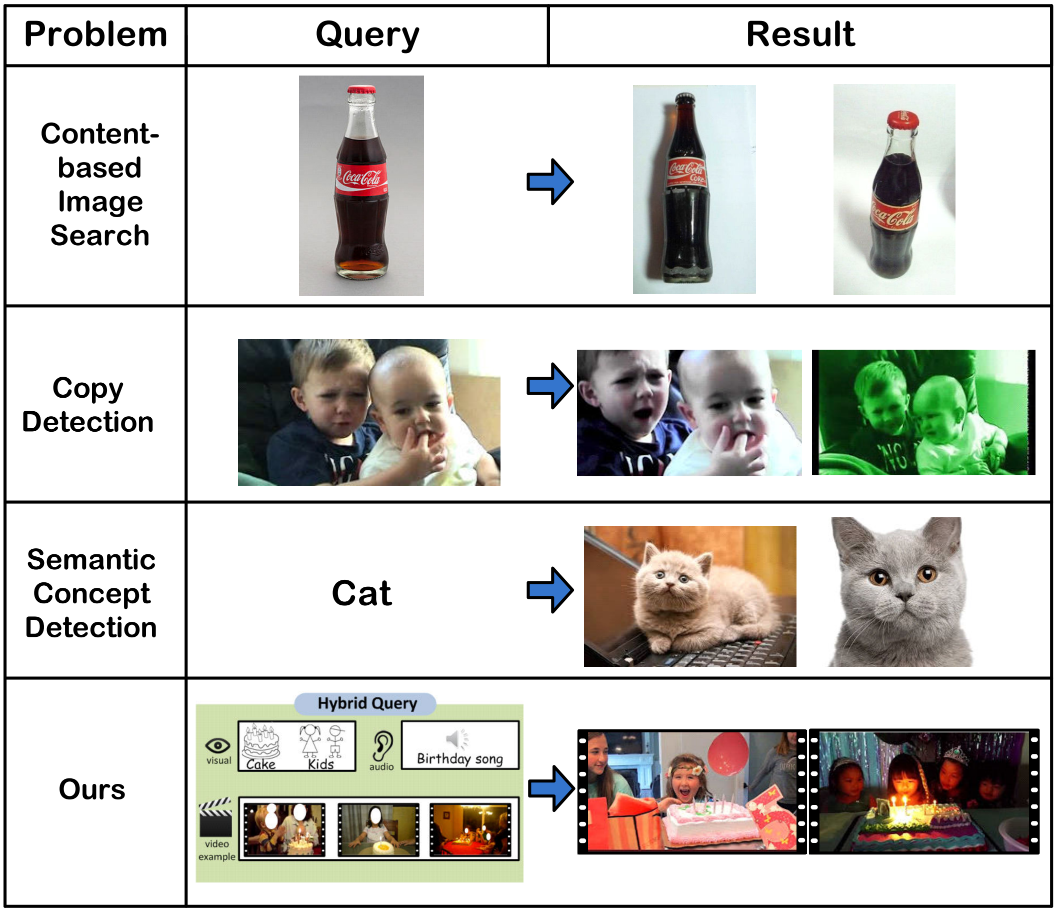


FIGURE 2.1: Comparison with different related problems.

called Video Google [32]. The system can be used to retrieve similar video key frames for a query image.

The content-based image retrieval method only utilizes the low-level descriptors that carry little semantic meaning. It is able to retrieve an instance of object from local descriptors matching, without knowing what is the object in the image. Therefore, it is good at finding visually similar but not necessarily semantically similar images.

2.2 Copy Detection

Video copy detection, also known as near duplicate detection, is a problem to detect whether a segment of video clip is derived from another video, typically by the means of various transformations such as addition, deletion, modification (of aspect, color, contrast or encoding) camcording, etc [5]. The query to copy detection is a video segment called copy, and the results are modified clips of the exact same video in a large video collection. See Fig. 2.1 for an example. This problem seems to be easier than the content-based image retrieval problem since the query and the retrieved results are essentially from the same video without significant changes.

Copy detection is also a well-studied problem. It has been broadly used to catch pirated copies of movies or copyrighted videos. Generally, the method relies on low-level visual/acoustic features without the need of understanding about the video content. The handcrafted or learned low-level features are used to generate a signature to search for the near-duplicate videos in a large collection. For example, Chum et al. [33] proposed to build min-hashing over the local descriptor SIFT to search for near duplicate images. Wu et al. [34] proposed a hierarchical approach for near-duplicate video detection. In this method, videos are first screened efficiently by global descriptors, and the remaining confusing videos will be inspected by an expensive approach that compares the local descriptors.

Similar to content-based image retrieval, copy detection only utilizes low-level descriptors that carry little semantic information. However, due to the natural of the problem, i.e. no semantic information is needed to find the near duplicate video, low-level features seems to be sufficient to solve the problem.

2.3 Semantic Concept Detection

Semantic concept detection, or concept detection for short, is a problem for searching the occurrence of a single concept in the video. As defined in Chapter 1, a concept is a visual or acoustic automatical tag of people, objects, scenes, actions, etc. over the video content [35]. The input of semantic concept detection is a text phrase, and the outputs are the videos that contain the corresponding concept. Note the search is purely based on the image or the video content. See Fig. 2.1. The challenge of semantic concept learning is to train a large number of robust classifiers called detectors on very big, and often noisy, datasets.

This line of study emerged in a TRECVID task called Semantic Indexing [36] in 2004 by NIST. In the image domain, a representative task is ImageNet [37] which was initiated in 2009. Concept detection is a general problem which includes action recognition [38–42] and scene recognition [43, 44]. It was initially targeted to detect people, objects or scenes in the news video [45]. News videos were selected as they are well-edited professional videos of limited variations. Later, it has been applied to "in-the-wide" challenging Internet videos, which are armature videos of low-resolution and significant camera motions.

Though the outputs of concept detector are high-level semantic features, the input of concept detectors are either low-level features or raw pixels. Studies on concept detection focused on learning better feature representation of images and videos. Initially it was achieved by handcrafted features such as SIFT and dense trajectories [46]. With the thrive of deep learning, the best feature representation is achieved through convolutional neural networks [47]. More recently, studies start to focus on learning detectors on noisy labeled web videos. Varadarajan et al. [48] discussed a method that exploits the YouTube topic API to train large scale video concept detectors on YouTube. The method utilized a calibration process and hard negative mining to train a second order mixture of experts model in order to discover correlations within the labels. Liang et al. [49] introduced a robust learning concept learning method that is inspired by the self-paced learning

Semantic concept detection is a simplified version of our problem, where it assumes the query is pure text and is only about a single concept. When the concept is not in the concept vocabulary, it will be detected by a set of combined in-vocabulary concepts. This problem of detecting is called zero-shot learning.

2.4 Multimedia Event Detection

With the advance in concept detection, people started to focus on addressing more complex queries called events. An event is more complex than a concept as it usually involves people engaged in process-driven actions with other people and/or objects at a specific place and time [21]. For example, the event "rock climbing" involves a climber, mountain scenes, and the action climbing. The relevant videos can be much diverse which may include videos about outdoor bouldering, indoor artificial wall climbing or snow mountain climbing.

A benchmark task on this topic is called TRECVID Multimedia Event Detection (MED) [50], which was initiated by NIST in 2010. Its goal is to assemble core detection technologies into a system that can search multimedia recordings for user-defined events based on pre-computed features [1]. In MED, users can search an event using either a few video examples (video-to-video) or a text description (text-to-video). MED is a challenging problem, and many studies have been published to address the problem. Generally, in a state-of-the-art video-to-video search system, the event classifiers are trained on the labeled videos using low-level and high-level features, and the final decision is derived from the fusion of the individual classification results. For example, Yu et al. [18] introduced an award-wining solution to address vide-to-video search problems. Later, the authors extended the video PQ encoding method for searching 1 million videos [51]. Oh et al. [52] presented a latent SVM event detector that enables for temporal evidence localization.

[1] https://www.nist.gov/itl/iad/mig/med-2016-evaluation

On the other hand, users can search an event using a text description. This text-to-video search resembles more of a real world scenario, in which users often start the search without any examples. As opposed to training an video-to-video event detector, it searches semantic concepts that are expected to occur in the relevant videos, e.g. we might look for concepts like “car”, “bicycle”, “hand” and “tire” for the event “changing a vehicle tire”. A few studies have been proposed on this topic [15, 53–57]. A closely related work is detailed in [58], where the authors presented their lessons and observations in building a state-of-the-art semantic search engine for Internet videos. More related studies can be found in Chapter 4.

2.5 Content-based Video Semantic Search

Our problem is similar to MED but advances it in the following perspectives. First, the queries are more complex which consist of both the text description of semantic features and a few video examples. As shown in Example 1.1, the query to our problem may contain semantic concepts, speech, visible texts, video examples, or any combination of these elements. Second, the search is performed solely based on high-level features about the content, as opposed to the low-level feature. As a result, the problem scale dealt in this thesis is orders-of-magnitude larger than that in the MED. This claim have been substantiated by the experiments in this thesis. For example, the biggest collection in TRECVID MED only contains around 200 thousand videos [5], whereas the largest dataset in this thesis contains 100 million videos.

Chapter 3

Indexing Semantic Features

3.1 Introduction

Semantic indexing aims at extracting semantic features in the video content and indexing them for efficient online search. In this chapter, we introduce the method for extracting and indexing semantic features from the video content, focusing on adjusting and indexing semantic concepts.

We consider indexing four types of semantic features in this thesis: visual concepts, audio concepts ASR and OCR. ASR provides acoustic information about videos. It especially benefits finding clues in close-to-camera and narrative videos such as "town hall meeting" and "asking for directions". OCR captures the text characters in videos with low recall but high precision. The recognized characters are often not meaningful words but sometimes can be a clue for fine-grained detection, e.g. distinguishing videos about "baby shower" and "wedding shower". ASR and OCR are text features, and thus can be conveniently indexed by the standard inverted index. The automatically detected text words in ASR and OCR in a video, after some preprocessing, can be treated as text words in a document. The preprocessing includes creating a stop word list for ASR from the English stop word list. The stop word lists for ASR includes utterances like "uh", "you know", etc. For OCR, due to the noise in word detection, we need to remove the words that do not exist in the English vocabulary.

How to index semantic concepts is an open question. Existing methods index a video by the raw concept detection score that is dense and inconsistent [9, 15, 53–57]. This solution is mainly designed for analysis and search over a few thousand of videos, and cannot scale to big data collections required for real world applications. Even though a modern text retrieval system can already index and search over billions of text documents, the task is still very challenging for semantic video search. The main reason is that semantic concepts are quite different from the text words, and semantic concept indexing is still an understudied problem. Specifically, concepts are automatically extracted by detectors with limited accuracy. The raw detection score associated with each concept is inappropriate for indexing for two reasons. First, the distribution of the scores is dense, i.e. a video contains every concept with a non-zero detection score, which is analogous to a text document containing every word in the English vocabulary. The dense score distribution hinders effective inverted indexing and search. Second, the raw score may not capture the complex relations between concepts, e.g. a video may have a "puppy" but not a "dog". This type of inconsistency can lead to inaccurate search results.

To address this problem, we propose a novel step called concept adjustment that aims at producing video (and video shot) representations that tend to be consistent with the underlying concept representation. After adjustment, a video is represented by a few salient and consistent concepts that can be efficiently indexed by the inverted index. In theory, the proposed adjustment model is a general optimization framework that incorporates existing techniques as special cases. In practice, as demonstrated in our experiments, the adjustment increases the consistency with the ground-truth concept representation on the real world TRECVID dataset. Unlike text words, semantic concepts are associated with scores that indicate how confidently they are detected. We propose an extended inverted index structure that incorporates the real-valued detection scores and supports complex queries with Boolean and temporal operators.

Compared to existing methods, the proposed method exhibits the following three benefits. First, it advances the text retrieval method for video retrieval. Therefore, while existing methods fail as the size of the data grows, our method is scalable, extending the current capability of semantic search by a few orders of magnitude while maintaining state-of-the-art performance. Our experiments validate this argument. Second, we propose a novel component called concept adjustment in a common optimization framework with solid probabilistic interpretations. Finally, our empirical studies shed some light on the tradeoff between efficiency and accuracy in a large-scale video search system. These observations will be helpful in guiding the design of future systems on related tasks.

The experimental results are promising on three datasets. On the TRECVID Multimedia Event Detection (MED), our method achieves comparable performance to state-of-the-art systems, while reducing its index by a relative 97%. The results on the TRECVID Semantic Indexing dataset demonstrate that the proposed adjustment model is able to generate more accurate concept representation than baseline methods. The results on the largest public multimedia dataset called YCCC100M [59] show that the method is capable of indexing and searching over a large-scale video collection of 100 million Internet videos. It only takes 0.2 seconds on a single CPU core to search a collection of 100 million Internet videos. Notably, the proposed method with reranking is able to achieve by far the best result on the TRECVID MED 0Ex task, one of the most representative and challenging tasks for semantic search in video.

3.2 Related Work

With the advance in object and action detection, people started to focus on searching more complex queries called events. An event is more complex than a concept as it usually involves people engaged in process-driven actions with other people and/or objects at a specific place and time [21]. For example, the event "rock climbing" involves video clips such as outdoor bouldering, indoor artificial wall climbing or snow mountain climbing. A benchmark task on this topic is called TRECVID Multimedia Event Detection (MED). Its goal is to detect the occurrence of a main event occurring in a video clip without any user-generated metadata. MED is divided into two scenarios in terms of whether example videos are provided. When example videos are given, a state-of-the-art system first train classifiers using multiple features and fuse the decision of the individual classification results [52, 60–67].

This thesis focuses on the other scenario named zero-example search (0Ex) where no example videos are given. 0Ex mostly resembles a real world scenario, in which users start the search without any example. As opposed to training an event detector, 0Ex searches semantic concepts that are expected to occur in the relevant videos, e.g. we might look for concepts like "car", "bicycle", "hand" and "tire" for the event "changing a vehicle tire". A few studies have been proposed on this topic [15, 53–57]. A closely related work is detailed in [58], where the authors presented their lessons and observations in building a state-of-the-art semantic search engine for Internet videos. Existing solutions are promising but only for a few thousand videos because they cannot scale to big data collections. Therefore, the biggest collection in existing studies contains no more than 200 thousand videos [5, 58].

Deng et al. [68] recently introduced label relation graphs called Hierarchy and Exclusion (HEX) graphs. The idea is to infer a representation that maximizes the likelihood and do not violate the label relation defined in the HEX graph.

3.3 Method Overview

Fig. 3.1 illustrates the proposed indexing stage for semantic concepts, there are four major components in this pipeline, namely, low-level feature extraction, concept detection, concept adjustment and inverted indexing, in which the concept adjustment component is first proposed in this thesis.

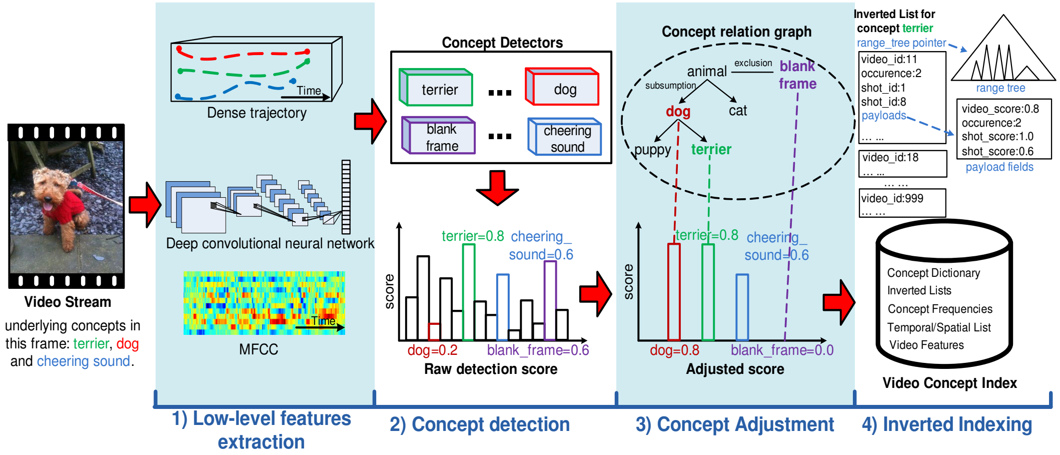


Figure 3.1: Pipeline for the semantic concept indexing.

A video clip is first represented by low-level visual or audio features. Common features include dense trajectories [16], deep learning [47] and MFCC features. The low-level features are then fed into the off-the-shelf detectors to extract the semantic concept features, in which each dimension corresponds to a confidence score of detecting a semantic (audio and visual) concept in a video shot. The dimensionality is equal to the number of unique detectors in the system.

We found that raw concept detection scores are inappropriate for indexing for two reasons: *distributional inconsistency* and *logical inconsistency*. The *distributional inconsistency* means that the distribution of the raw detection score is inconsistent with the underlying concept distribution of the video. The underlying concept representation tends to be sparse but the distribution of the detection score is dense, i.e. a video contains every concept. Indexing the dense representation by either dense matrices or inverted indexes is known to be inefficient. For example, Fig. 3.1 illustrates an example in which the raw concept detection contains 14 non-zero scores but there are only three concepts in the underlying representation: “dog”, “terrier”, and “cheering sound”. As we see, the dense distribution of the raw detection score is very different from the underlying distribution.

The *logical inconsistency* means that the detection scores are not consistent with the semantic relation between concepts, e.g. a video contains a “terrier” but not a “dog”. This type of inconsistency results from that 1) the detectors are usually trained by different people using different data, features and models. It is less likely for them to consider the concept consistency that is not in their vocabulary; 2) even within a concept vocabulary, many classification models cannot capture complex relation between concepts [68]. The inconsistent representation can lead to inaccurate search results if not properly handled. For example, in Fig. 3.1, the score of “dog” 0.2 is less than the score of “terrier” 0.8; the frame is detected as “blank frame”, which means a empty frame, and a “terrier”.

To address the problem of distributional and logical inconsistencies, we propose a novel step called concept adjustment. It aims at generating consistent concept representations that can be efficiently indexed and searched. We propose an adjustment method based on the recently proposed label relation graph [68] that models the hierarchy and exclusion relation between concepts (see Step 3 in Fig. 3.1). After adjustment, a video is represented by a few salient concepts, which can be efficiently indexed by the inverted index. In addition, the adjusted representation is logically consistent with the complex relation between concepts.

An effective approach is to index the adjusted representation by the inverted index in text retrieval. However, unlike text words, semantic concepts are associated with scores that indicate how confidently they are detected. The detection score cannot be directly indexed by the standard inverted index. As a result, the scores are usually indexed by dense matrices in existing methods [55, 58]. To this end, we modify the inverted index structure so that it can index the real-valued adjusted score. The modified index contains inverted indexes, frequency lists that store the concept statistics used in the retrieval model, temporal lists that contain the shot information, and video feature lists that store the low-level features. The extended index structure is compatible to existing text retrieval algorithms.

3.4 Concept Adjustment

In this thesis, we make no assumption on the training process of the off-the-shelf concept detectors. The detectors may be trained by any type of features, models or data. We relax the assumption [68] that detectors must be "re-trainable" by particular training algorithms because this is usually impossible when we do not have the access to the training data, the code or the computational resource.

Concept adjustment aims at generating video (or video shot) representations that tend to be consistent to the underlying concept representation and meanwhile can be searched efficiently. An ideal video representation tends to be similar to the underlying concept representation in terms of the distributional and logical consistency. To this end, we propose an optimization model to find consistent video representations of the given raw concept detection output. Formally, let $\mathbf{D} \in \mathbb{R}^{n \times m}$ denote the raw scores outputted by the concept detectors, where the row represents the n shots in a video, and the column represents the m visual/audio concepts. The prediction score of each concept is in the range between 0 and 1, i.e. $\forall i, j, \mathbf{D}_{ij} \in [0, 1]$. We are interested in obtaining a consistent representation $\mathbf{v} \in \mathbb{R}^{m \times 1}$, which can be obtained by solving the following optimization problem:

$$\begin{aligned} &\mathop{\arg\min}_{\mathbf{v}\in[0,1]^m} \frac{1}{2}\|\mathbf{v} - f_p(\mathbf{D})\|_2^2 + g(\mathbf{v};\alpha,\beta) \\ &\qquad\qquad \text{subject to } \mathbf{A}\mathbf{v} \le \mathbf{c} \end{aligned} \tag{3.1}$$

where

$$f_p(\mathbf{D}) = (1 - (\frac{m-1}{m})^p)[\|\mathbf{d}_1\|_p, \dots, \|\mathbf{d}_m\|_p]^T \tag{3.2}$$

and $\mathbf{D} = \begin{bmatrix} | & & | \\ \mathbf{d}_1 & \cdots & \mathbf{d}_m \\ | & & | \end{bmatrix}$. Each element of $f_p(\mathbf{D})$ is the p-norm of the column vector of $\mathbf{D}$. $g(\mathbf{v};\alpha,\beta)$ is a regularizer of $\mathbf{v}$ with the parameters α and β. $g(\cdot)$ imposes the distributional consistency, and will be discussed in Section 3.4.1. $\mathbf{A}$ and $\mathbf{c}$ are a constant matrix and a constant vector, which model the logical consistencies and will be discussed in Section 3.4.2. It is easy to verify that when $p = \infty$ and $p = 1$, the operator $f_p(\mathbf{D})$ corresponds to the max and the average pooling operator. Usually $g(\cdot)$ is convex, and thus Eq. (3.1) can be conveniently solved by the standard convex programming toolbox [69]. The raw prediction score may diminish during the concept adjustment. It is usually helpful to normalize the optimal value of $\mathbf{v} = [v_1, \cdots, v_m]$ by:

$$\hat{v}_i = \min(1, \frac{v_i}{\sum_{j=1}^m v_j} \sum_{j=1}^m f_p(\mathbf{D})_j I(v_j)), \tag{3.3}$$

where $\hat{\mathbf{v}} = [\hat{v}_1, \cdots, \hat{v}_m]$ is the adjusted score after normalization. $I(v)$ is an indicator function equalling 1 when $v > 0$, and 0 otherwise. Here we define $0/0 = 0$.

In order to obtain the shot-level adjusted representation, we can treat a shot as a “video” and let $\mathbf{D}$ be a single row matrix containing the detection score of the shot. Eq. (3.1) can be used but with an extra integer set in the constraints (see Section 3.4.2).

3.4.1 Distributional Consistency

For the distributional consistency, a regularization term $g(\mathbf{v};\alpha,\beta)$ is introduced that produces sparse representations while taking into account that certain concepts may co-occur together. A naive implementation is to use the l_0 norm:

$$g(\mathbf{v};\alpha,\beta) = \frac{1}{2}\beta^2\|v\|_0. \tag{3.4}$$

This regularization term presents a formidable computational challenge. In this thesis we propose a more feasible and general regularization term. Suppose the concepts are divided into q non-overlapping groups. A group may contain a number of co-occurring concepts, or a single concept if it does not co-occur with others. Such sparsity and group

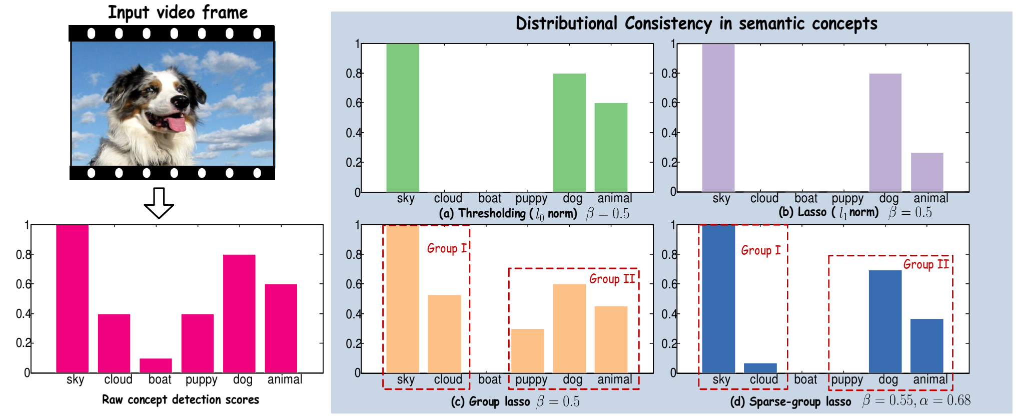


FIGURE 3.2: Comparison of different regularization terms.

sparsity information can be encoded into the model by adding a convex regularization term $g(\mathbf{v})$ of the l_1 norm and the sum of group-wise l_2 norm of $\mathbf{v}$:

$$g(\mathbf{v};\alpha,\beta) = \alpha\beta\|\mathbf{v}\|_1 + (1-\alpha)\sum_{l=1}^{q}\beta\sqrt{p_l}\|\mathbf{v}^{(l)}\|_2, \tag{3.5}$$

where $\mathbf{v}^{(l)} \in \mathbb{R}^{p_l}$ is the coefficient for the lth group where p_l is the length of that group. $\alpha \in [0,1]$ and β are two parameters controlling the magnitude of the sparsity.

The parameter α balances the group-wise and the within-group sparsity. When $\alpha = 1$, $g(\mathbf{v})$ becomes *lasso* [70] that finds a solution with few nonzero entries. When $\alpha = 0$, $g(\mathbf{v})$ becomes *group lasso* [71], that only yields nonzero entries in a sparse set of groups. If a group is included then all coefficients in the group will be nonzero. Sometimes, the sparsity within a group is also needed, i.e. if a group is included, only few coefficients in the group will be nonzero. This is known as *sparse-group lasso* [72] that linearly interpolates *lasso* and *group lasso* by the parameter α.

In the context of semantic concepts, *lasso* is an approximation to the corresponding l_0 norm regularization problem which is computationally expensive to solve. *Lasso* and the l_0 norm term assume the concepts are independent, and works well when the assumption is satisfied. On the other hand, *Group lasso* assumes the there exist groups of concepts that tend to be present or absent together frequently, e.g. “sky/cloud”, “beach/ocean/waterfront” and “table/chair”. The group may also include multimodal concepts such as “baby/baby noises”. Since co-occurring concepts may not always be present together, the within-group sparse solution is needed sometimes, i.e. only few concepts in a group are nonzero. This can be satisfied by *sparse-group lasso* that makes weaker assumptions about the underlying concept distribution.

Consider a toy example in Fig. 3.2 comparing the above regularization terms. The raw prediction scores of the input frame is shown below the video frame, where the ground truth concept representation contains only “sky”, “cloud”, “dog” and “animal”. The figures on the right illustrate the optimal solutions yielded by Eq. (3.5) after normalization with l_0 norm, l_1 norm, group lasso and sparse-group lasso. As we see, l_0 and l_1 norm do not consider the co-occurring concepts, and the concept "cloud" is missing in their solutions (Fig. 3.2 (a) and (b)). In contrast, the concept "cloud" in the solution of group lasso and sparse-group lasso is recalled by the concept group "sky" and "cloud" (Fig. 3.2 (c) and (d)). The optimal solution of group lasso, however, introduces a false positive concept "puppy", because when it selects a group, all coefficients in the group will be nonzero. However, only few concepts in a group are nonzero in the solution of sparse-group lasso. See Fig. 3.2 (d).

3.4.2 Logical Consistency

The concept relation is modeled by Hierarchy and Exclusion (HEX) graph. Following Deng et al. [68], we assume that the graph is given beforehand. According to [68], a HEX graph is defined as:

Definition 3.1. A HEX graph $G = (N, E_h, E_e)$ is a graph consisting of a set of nodes $N = \{n_1, \cdots, n_m\}$, directed edges $E_h \subseteq N \times N$ and undirected edges $E_e \subseteq N \times N$ such that the subgraph $G_h = (N, E_h)$ is a directed acyclic graph and the subgraph $G_e = (N, E_e)$ has no self-loop.

Each node in the graph represents a distinct concept. A hierarchy edge $(n_i, n_j) \in E_h$ indicates that concept n_i subsumes concept n_j in the concept hierarchy, e.g. "dog" is a parent of "puppy". An exclusion edge $(n_i, n_j) \in E_e$ indicates concept n_i and n_j are mutually exclusive, e.g. a frame cannot be both "blank frame" and "dog". Based on Definition 3.1, we define the logically consistent representation as:

Definition 3.2. $\mathbf{v} = [v_1, \cdots, v_m]$ is a vector of concept detection scores. The ith dimension corresponds to the concept node $n_i \in N$ in the HEX graph G. $\mathbf{v} \in [0,1]^m$ is logically consistent with G if for any pair of concepts (n_i, n_j):

1. if $n_i \in \alpha(n_j)$, then $v_i \geq v_j$;
2. if $\exists n_p \in \bar{\alpha}(n_i)$, $\exists n_q \in \bar{\alpha}(n_j)$ and $(n_p, n_q) \in E_e$, then we have $v_i v_j = 0$;

where $\alpha(n_i)$ is a set of all ancestors of n_i in G_h, and $\bar{\alpha}(n_i) = \alpha(n_i) \cup n_i$.

Definition 3.2 indicates that a logically consistent representation should not violate any concept relation defined in its HEX graph G. This definition generalizes the legal assignments in [68] to allow concepts taking real values. We model the logical consistency by the affine constraints $\mathbf{Av} \leq \mathbf{c}$. The constant matrix $\mathbf{A}$ and vector $\mathbf{c}$ can be calculated from Algorithm 1. For each edge in the graph, Algorithm 1 defines a constraint on values the two concepts can take. A hierarchy edge $(n_i, n_j) \in E_h$ means that the value of a parent is no less than the value of its children, e.g. "puppy=0.8" but "dog=0.2" is inconsistent. For each exclusion edge, Algorithm 1 introduces an affine constraint $v_i + v_j = 1$ and $v_i, v_j \in \{0, 1\}$ to avoid the case where two concepts both have nonzero values. Note that the solution of the exclusion constraint complies with the binary legal assignments in [68] that for any $(n_i, n_j) \in E_e$, $(v_i, v_j) \neq (1, 1)$. According to Definition 3.2, it is easy to prove that the optimal solution of Eq. (3.1) is logically consistent with a given HEX graph. The problem with integer constraints can be solved either by the mixed-integer convex programming toolbox, or by the constraint relaxation [73].

Theorem 3.3. *The optimal solutions of Eq.* (3.1) *(before or after normalization) is logically consistent with its given HEX graph.*

Algorithm 1: Constraints for logical consistency.

```
input  : A HEX graph $G = (V, E_h, E_e)$
output: A constant matrix $\mathbf{A}$ and a constant $\mathbf{c}$.
$n = |E_h| + |E_e|$; $m = |V|$; $k = 0$;
$\mathbf{A} = \mathbf{0}_{n \times m}$, $\mathbf{c} = \mathbf{0}_{n \times 1}$;
foreach $(n_i, n_j) \in E_h$ do
    $\mathbf{A}_{ki} = -1$; $\mathbf{A}_{kj} = 1$; $\mathbf{c}_k = 0$;
    $k{+}{+}$;
end
Define an integer constraint set $\mathbb{I} \leftarrow \phi$;
foreach $(n_i, n_j) \in E_e$ do
    $\mathbf{A}_{ki} = 1$; $\mathbf{A}_{kj} = 1$; $\mathbf{c}_k = 1$;
    add $n_i$, $n_j$ to $\mathbb{I}$;
    $k{+}{+}$;
end
return $\mathbf{A}$, $\mathbf{c}$, $\mathbb{I}$;
```

3.4.3 Discussions

The proposed model can produce a representation that tends to be both distributionally and logically consistent to the underlying concept representation. A nice property of the model in Eq. (3.1) is that it can degenerate to several existing methods. For example, it is easy to verify that the max and the average pooling results are optimal solutions of Eq. (3.1) in special cases. Theorem 3.3 indicates that the optimal solution of adjusted representations complies with the logical consistency definition. Theorem 3.4 indicates that the thresholding and the top-k thresholding results are optimal solutions of Eq. (3.1) in special cases. The thresholding method preserves scores only above some threshold. In some cases, instead of using an absolute threshold, one can alternatively set the threshold in terms of the number of concepts to be included. This is known as the top-k thresholding. See the proof in Appendix A.

Theorem 3.4. *The thresholding and the top-k thresholding results are optimal solutions of Eq.* (3.1) *in special cases.*

The choice of the proposed model parameters depends on the underlying distribution of the semantic concepts. For the manually exclusive concepts, such as the 1,000 concepts in the ImageNet challenge [74], the l_0 norm or the l_1 norm without any HEX constraint should work reasonably well. In addition, as the model is simple, the problem can be efficiently solved by the closed-form solution. When the concepts are of concrete hierarchical or exclusion relations, such as the concepts in TRECVID SIN [36], incorporating the HEX constraint tends to be beneficial. The group-lasso and the sparse-group lasso play a role when groups of concepts tend to co-occur together frequently. It can be important for the multimodal concept detectors that capture the same concept by multiple features, e.g. audio or visual. An approach to derive the co-occurring concepts is by clustering the concepts in their labeled training data. We observed big clusters tend to include loosely coupled concepts, e.g. sky/cloud is a good group, but sky/cloud/helicopter is not. To be prudent, we recommend limiting the group size in clustering.

Note the exclusion relation between concepts only makes sense at the shot-level adjustment, as in the video-level representation the scores of exclusive concepts can be both non-zeros. Solving a mixed integer convex programming problem takes more time than solving a regular convex programming problem. So when the proposed method is applied on shot-level features, it is useful to use some type of constraint relaxation techniques. Besides, in the current model, we assume the concept detectors are equally accurate. A simple extension to embed this information is by discounting the squared loss of inaccurate concepts in Eq. (3.1).

The proposed model also provides common interpretations of what are being optimized. The physical meaning of the optimization problem in Eq. (3.1) can be interpreted as a maximum a priori model. The physical meaning of the optimization problem in Eq. (3.1) can be interpreted as the following. Firstly, for the pooling function $f_p(\mathbf{D})$ defined on concept features $\mathbf{D}$, the average pooling ($p = 1$) and max pooling ($p = \infty$) correspond to the maximum likelihood estimation (MLE) for the mean and the maximum of the concept feature variable under the assumption that the utilized data are i.i.d. sampled from its latent distribution, respectively. This actually regularizes which feature should be specified to deliver the major information underlying a video. In specific, the average pooling emphasizes that the main information is uniformly scattered over the video, while max pooling underlines that the most dominant feature should be the most representative feature.

We then give a probabilistic explanation for the model in Eq. (3.1). As a general maximum a priori (MAP) model, it consists of likelihood and prior elements. The likelihood term is constructed based on the assumption that the observation $f_p(\mathbf{D})$ is approximated by the latent term $\mathbf{v}$ under Gaussian corruptions, i.e. $f_p(\mathbf{D}) = \mathbf{v} + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2 \mathbf{I}_n)$. This naturally leads to the likelihood term:

$$p(f_p(\mathbf{D})|\mathbf{v}) = N(\mathbf{v}, \sigma^2 \mathbf{I}_n). \tag{3.6}$$

We utilize two useful priors. The first is on the logical consistency. That is, meaningful concept features should be located in the feasible region induced by the constraint $\Psi = \{\mathbf{v} | A\mathbf{v} \leq \mathbf{c}\}$. This naturally leads to the following prior distribution for $\mathbf{v}$:

$$\phi(\mathbf{v}) = \begin{cases} \gamma, & \mathbf{v} \in [0,1]^m \cap \Psi \\ 0, & \text{otherwise} \end{cases}, \tag{3.7}$$

where $\gamma = 1/|\Psi| = 1/\int_\Psi d\mathbf{v}$ is the normalization factor. Another prior is induced by the sparsity of concepts in a video, which can be encoded as the following prior distribution:

$$\pi(\mathbf{v}) \propto \exp\left\{-\alpha\beta \|\mathbf{v}\|_1 - (1-\alpha)\sum_{l=1}^{q} \beta\sqrt{p_l}\|\mathbf{v}^{(l)}\|_2\right\}. \tag{3.8}$$

This corresponds to the sparse group lasso estimator [75].

Based on Bayes theorem, we can then get the posterior distribution for $\mathbf{v}$ as:

$$p(\mathbf{v}|f_p(\mathbf{D})) \propto p(f_p(\mathbf{D})|\mathbf{v})\pi(\mathbf{v})\phi(\mathbf{v}), \tag{3.9}$$

and can then determine $\mathbf{v}$ by maximizing this posterior (i.e., the MAP rule). This complies with the concept adjustment optimization model in Eq. (3.1) in the paper, and thus provides a Bayesian understanding for this deterministic problem.

3.5 Inverted Indexing & Search

After adjustment, a video is represented by a few salient and consistent concepts. In analogy to words in a text document, concepts can be treated as "words" in a video. Unlike text words, concepts are associated with scores that indicate how confidently they are detected. The real-valued scores are difficult to be directly indexed in the standard inverted index designed for text words. A naive approach is by binning, where we assign real values to the bins representing the segment covering the numerical value. The concepts are duplicated by the number of its filled bins. However, this solution creates hallucinating concepts in the index, and cannot store the shot-level concept scores.

To solve the problem in a more principled way, we propose a modified inverted index to incorporate the real-valued detection scores. In text retrieval, each unique text word has a list of *postings* in the inverted index. A *posting* contains a document ID, the term frequency, and the term positions in the document. The term frequency is used in the retrieval model, and the position information is used in the proximity search. Inspired by this structure, in our system, the concept with a nonzero score in the adjusted representation is indexed to represent a video. Each unique concept has a list of *video postings* and a range search tree. An example index structure is illustrated in Step 4 in Fig. 3.1. A *video posting* contains a video ID, the number of concept occurrence in the video, a video-level detection score, and a list of video shots in which the concept occurs. It also has a payload to store the real-valued detection score for each shot. In this way, the query that searches for the video-level score of a certain range can be handled by the range tree, e.g. "videos that contain dog > 0.5"; the query that searches for the shot-level score can be handled by the payload in the posting, e.g. "shots that contain dog > 0.5"; otherwise, the query can be processed in a similar way as in text retrieval using the adjusted video-level score, e.g. videos that contain "dog AND cat".

3.5.1 Video Search

A search usually contains two steps: retrieving a list of *video postings* and ranking the postings according to some retrieval model. In our system, we consider the following query operators to retrieve a *video posting* list:

- **Modality query**: Searching a query term in a specified modality. For example, "visual:dog" returns the videos that contain the visual concept "dog"; "visual:dog/[score s_1, s_2]" returns the videos that have a detection score of "dog" between s_1 and s_2. "visual" is the default modality. The other modalities are "asr" for automatically recognized speech, "ocr" for recognized optical characters, and "audio" for audio concepts.
- **Temporal query**: Searching query terms that have constraints on their temporal occurrences in a video. The constraints can be specified in terms of the absolute timestamp like "videos that contain dog between the time t_1 and t_2", the relative sequence like "videos in which dog is seen before cat", or the proximity relations like "videos that contain dog and cat within the time window of t_1". A temporal query can be handled in a similar fashion as the proximity search in text search.
- **Boolean query**: Multiple terms can be combined together with Boolean operators to form a more complex query. Our system supports three operators: "AND", "OR" and "AND NOT", where the "OR" operator is the default conjunction operator.

A Boolean query can be handled by the standard algorithms in text retrieval, as Theorem 3.2 guarantees that the adjusted representation is logically consistent. However, the query may be accelerated by utilizing the concept relation in the HEX graph. For example, it is unnecessary to run a query to realize that ("dog" AND "animal") = "dog".

Suppose the query is expressed in the disjunctive normal form. Given a HEX graph G and two concepts $n_i, n_j \in V$, for each term in the disjunctive normal form, we apply: $(n_i$ AND $n_j) = n_i$ if $n_j \in \alpha(n_i)$, where $\alpha(n_i)$ is the set of all ancestors of n_i in G_h; $(n_i$ AND NOT $n_j) = \phi$ if $\exists n_p \in \bar{\alpha}(n_i)$, $\exists n_q \in \bar{\alpha}(n_j)$ and $(n_p, n_q) \in E_e$. The simplified query can be then used to retrieval the *video postings*.

After retrieving a *video posting* list, the next step is to rank the postings according to some retrieval model. A retrieval model can have substantial impact on the performance. We will study the impact of retrieval models in the next chapter. For now we use the Okapi BM25 model [76]. Suppose the input query is $Q = q_1, \cdots, q_n$, the model ranks a video d by:

$$\mathrm{s}(d|Q) = \sum_{i=1}^{n} \log \frac{|C| - df(q_i) + \frac{1}{2}}{df(q_i) + \frac{1}{2}} \frac{tf(q_i, d)(k_1+1)}{tf(q_i, d) + k_1(1 - b + b\frac{len(d)}{\overline{len}})}, \tag{3.10}$$

where $|C|$ is the total number of videos. $tf(q_i, d)$ returns the score of the concept q_i in the adjusted representation of video d. $df(\cdot)$ calculates the sum of adjusted score of q_i in the video collection. $len(d)$ calculates the sum of adjusted scores for video d, and $\overline{len}$ is the average length across all videos. k_1 and b are two parameters to tune. The statistics are calculated by the adjusted concept score rather than the raw detection score.

3.6 Experiments

3.6.1 Setups

Dataset and evaluation: The experiments are conducted on two TRECVID benchmarks called Multimedia Event Detection (MED): MED13Test and MED14Test [5]. The performance is evaluated by several metrics for a better understanding, which include: P@20, Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and MAP@20, where the MAP is the official metric used by NIST. Each set includes 20 events over 25,000 test videos. The official NIST's test split is used. We also evaluate each experiment on 10 randomly generated splits to reduce the split partition bias. All experiments are conducted without using any example or text metadata.

Features and queries: Videos are indexed by semantic features including semantic visual concepts, ASR, and OCR. For semantic concepts, 1,000 ImageNet concepts are trained by the deep convolution neural networks [47]. The remaining 3,000+ concepts are directly trained on videos by the self-paced learning pipeline [77, 78] on around 2 million videos using improved dense trajectories [16]. The video datasets include Sports [79], Yahoo Flickr Creative Common (YFCC100M) [59], Internet Archive Creative Common

Table 3.1: Summary of the semantic concept training sets. ImageNet features are trained on still images, and the rest are trained on videos.

Dataset	#Samples	#Classes	Category	Example Concepts
DIY [80]	72,000	1,601	Instructional videos	Yoga, Juggling, Cooking
IACC [5]	600,000	346	Internet archive videos	Baby, Outdoor, Sitting down
YFCC100M [59]	800,000	609	Amateur videos on Flickr	Beach, Snow, Dancing
ImageNet [37]	1,000,000	1000	Still images	Bee, Corkscrew, Cloak
Sports [79]	1,100,000	487	Sports videos on YouTube	Bullfighting, Cycling, Skiing

(IACC) [5] and Do It Yourself (DIY) [80]. The details of these datasets can be found in Table 3.1. The ASR module is built on EESEN and Kaldi [19, 20, 81]. OCR is extracted by a commercial toolkit. Three sets of queries are used: 1) *Expert* queries are obtained by human experts; 2) *Auto* queries are automatically generated by the Semantic Query Generation (SQG) methods in [58] using ASR, OCR and visual concepts; 3) *AutoVisual* queries are also automatically generated but only includes the visual concepts. The *Expert* queries are used by default.

Configurations: The concept relation released by NIST is used to build the HEX graph for IACC features [35][1]. The adjustment is conducted at the video-level average ($p = 1$ in Eq. (3.1)) so no shot-level exclusion relations are used. For other concept features, since there is no public concept relation specification, we manually create the HEX graph. The HEX graphs are empty for Sports and ImageNet features as there is no evident hierarchical and exclusion relation in their concepts. We cluster the concepts based on the correlation of their training labels, and include concepts that frequently co-occur together into a group. The parameters are tuned on a validation sets, and then are fixed across all experiment datasets including MED13Test, MED14Test and YFCC100M. Specifically, the default parameters in Eq. (3.1) are $p = 1$, $\alpha = 0.95$. β is set as the top k detection scores in a video, and is different for each type of features: 60 for IACC, 10 for Sports, 50 for YFCC100M, 15 for ImageNet, and 10 for DIY features. CVX optimization toolbox [69] is used to solve the model in Eq. (3.1). Eq. (3.10) is used as the retrieval model for concept features, where $k_1 = 1.2$ and $b = 0.75$.

3.6.2 Performance on MED

We first examine the overall performance of the proposed method. Table 3.2 lists the evaluation metrics over the two benchmarks on the standard NIST split and on the 10 randomly generated splits. The performance is reported over three set of queries: *Expert*, *Auto*, and *AutoVisual*.

Table 3.3 compares the performance of the raw and the adjusted representation on the 10 splits of MED13Test. *Raw* lists the performance of indexing the raw score by dense matrices; *Adjusted* lists the performance of indexing the adjusted concepts by

[1] http://www-nlpir.nist.gov/projects/tv2012/tv11.sin.relations.txt

the proposed index which preserves the real-valued scores. As we see, although *Raw* is slightly better than *Adjusted*, its index in the form of dense matrices is more than 33 times bigger than the inverted index in *Adjusted*. The comparison substantiates that the adjusted representation has comparable performances with the raw representation but can be indexed by a much smaller index.

An interesting observation is that *Adjusted* outperforms *Raw* on 8 out of 20 events on MED13Test (see Table B.1). We inspected the results and found that concept adjustment can generate more consistent representations. Fig. 3.3 illustrates raw and adjusted concepts on three example videos. Since the raw score is dense, we only list the top ranked concepts. As we see, the noisy concept in the raw detection may be removed by the logical consistency, e.g. "snow" in the first video. The missed concept may be recalled by logical consistencies, e.g. "vehicle" in the third video is recalled by "ground vehicle". The frequently co-occurring concepts may also be recovered by distributional consistencies, e.g. "cloud" and "sky" in the second video. Besides, we also found that Boolean queries can boost the performance. For example, in "E029: Winning a race without a vehicle", the query of relevant concepts such as swimming, racing or marathon can achieve an AP of 12.5. However, the Boolean query also containing "AND NOT" concepts such as car racing or horse riding can achieve an AP of 24.5.

Table 3.2: Overview of the system performance.

(a) Performance on the NIST's split

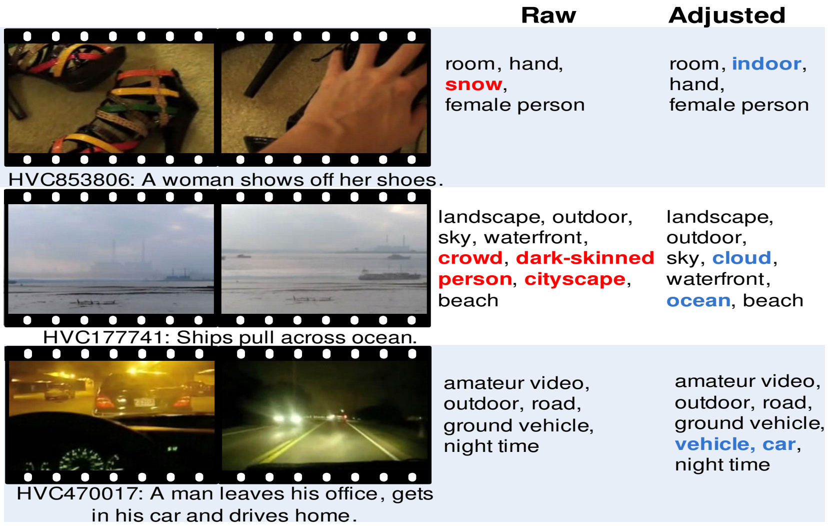
Dataset	Query	Evaluation Metric			
		P@20	MRR	MAP@20	MAP
MED13Test	Expert	0.355	0.693	0.280	0.183
	Auto	0.243	0.601	0.177	0.118
	AutoVisual	0.125	0.270	0.067	0.074
MED14Test	Expert	0.228	0.585	0.147	0.172
	Auto	0.150	0.431	0.102	0.100
	AutoVisual	0.120	0.372	0.067	0.086

(b) Average Performance on the 10 splits

Dataset	Query	Evaluation Metric			
		P@20	MRR	MAP@20	MAP
MED13Test	Expert	0.325	0.689	0.247	0.172
	Auto	0.253	0.592	0.187	0.120
	AutoVisual	0.126	0.252	0.069	0.074
MED14Test	Expert	0.219	0.540	0.144	0.171
	Auto	0.148	0.417	0.084	0.102
	AutoVisual	0.117	0.350	0.063	0.084

Table 3.3: Comparison of the raw and the adjusted representation on the 10 splits.

Method	Index	Evaluation Metric			
		P@20	MRR	MAP@20	MAP
MED13 Raw	385M	0.312	0.728	0.230	0.176
MED13 Adjusted	11.6M	0.325	0.689	0.247	0.172
MED14 Raw	357M	0.233	0.610	0.155	0.185
MED14 Adjusted	12M	0.219	0.540	0.144	0.171



	Raw	Adjusted
HVC853806: A woman shows off her shoes.	room, hand, **snow**, female person	room, **indoor**, hand, female person
HVC177741: Ships pull across ocean.	landscape, outdoor, sky, waterfront, **crowd**, **dark-skinned person**, **cityscape**, beach	landscape, outdoor, sky, **cloud**, waterfront, **ocean**, beach
HVC470017: A man leaves his office, gets in his car and drives home.	amateur video, outdoor, road, ground vehicle, night time	amateur video, outdoor, road, ground vehicle, **vehicle, car**, night time

FIGURE 3.3: Comparison of raw and adjusted concepts.

The parameters α and β in Eq. (3.1) control the magnitude of sparsity in the concept adjustment, i.e. the percentage of concepts with nonzero scores in a video representation. A sparse representation reduces the size of indexes but hurts the performance at the same time. As we will see later, β is more important than α in affecting the performance. Therefore, we fix α to 0.95 and study the impact of β. Fig. 3.4 illustrates the tradeoff between accuracy and efficiency on the 10 splits of MED13Test. By tuning β, we obtain different percentages of nonzero concepts in a video representation. The x-axis lists the percentage in the log scale. $x = 0$ indicates the performance of ASR and OCR without semantic concept features. We discovered that we do not need many concepts to index a video, and a few adjusted concepts already preserve significant amount of information for search. As we see, the best tradeoff in this problem is 4% of the total concepts (i.e. 163 concepts). Further increasing the number of concepts only leads to marginal performance gain.

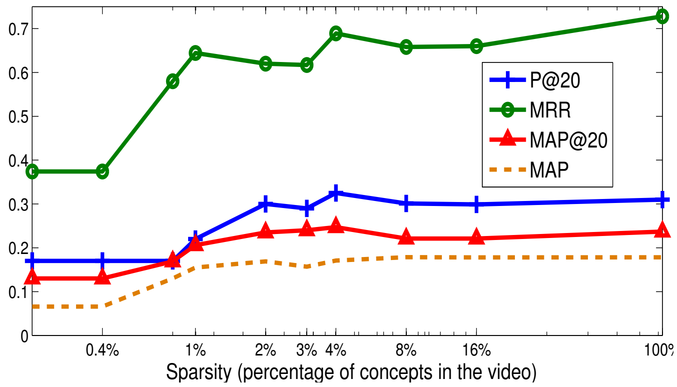


FIGURE 3.4: The impact of parameter β. $x = 0$ indicates the performance of ASR and OCR without semantic concepts.

3.6.3 Comparison to State-of-the-art on MED

We then compare our best result with the published results on MED13Test. The experiments are all conducted on the NIST's split, and thus are comparable to each other. As we see in Table 3.4, the proposed method has a comparable performance to the state-of-the-art methods. Notably, the proposed method with one iteration of reranking [15] is able to achieve the best result. The comparison substantiates that our method maintains state-of-the-art accuracy. It is worth emphasizing that the baseline methods may not scale to big data sets, as the dense matrices are used to index all raw detection scores [15, 55, 58].

Table 3.4: MAP (× 100) comparison with the published results on MED13Test.

Method	Year	MAP
Composite Concepts [53]	2014	6.4
Tag Propagation [54]	2014	9.6
MMPRF [56]	2014	10.1
Clauses [57]	2014	11.2
Multi-modal Fusion [55]	2014	12.6
SPaR [15]	2014	12.9
E-Lamp FullSys [58]	2015	20.7
Our System	2015	18.3
Our System + reranking	**2015**	**20.8**

3.6.4 Comparison to Top-k Thresholding on MED

We compare our full adjustment model with its special case top-k thresholding on MED14Test. Theorem 3.4 indicates that the top-k thresholding results are optimal solutions of our model in special cases. The experiments are conducted using IACC SIN346 concept features that have large HEX graphs. We select the features because large HEX graphs help compare the difference between the two methods. Table 3.5 lists the average performance across 20 queries. We set the parameter k (equivalently β) to be 50, and 60. As the experiment only uses 346 concepts, the results are worse than our full systems using 3000+ concepts.

Table 3.5: Comparison of the full adjustment model with its special case Top-k Thresholding on the 10 splits of MED14Test.

Method	k	Evaluation Metric			
		P@20	MRR	MAP@20	MAP
Our Model	50	**0.0392**	**0.137**	**0.0151**	**0.0225**
Top-k	50	0.0342	0.0986	0.0117	0.0218
Our Model	60	**0.0388**	**0.132**	**0.0158**	**0.0239**
Top-k	60	0.0310	0.103	0.0113	0.0220

As we see, the full adjustment model improves the accuracy and outperforms Top-k thresholding in terms of P@20, MRR and MAP@20. We inspected the results and found that the full adjustment model can generate more consistent representations (See Fig. 3.3). The results suggest that the full model outperforms the special model in this problem.

3.6.5 Accuracy of Concept Adjustment

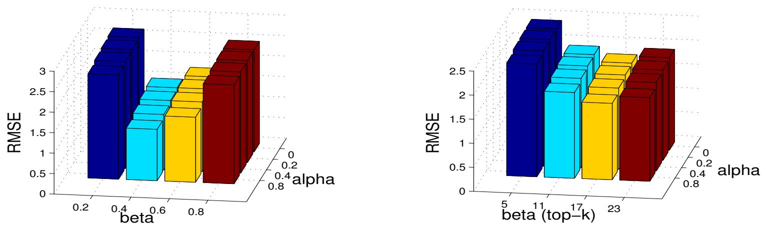
Generally the comparison in terms of retrieval performance depends on the query words. A query-independent way to verify the accuracy of the adjusted concept representation is by comparing it to the ground truth representation. To this end, we conduct experiments on the TRECVID Semantic Indexing (SIN) IACC set, where the manually labeled concepts are available for each shot in a video. We use our detectors to extract the raw shot-level detection score, and then apply the adjustment methods to obtain the adjusted representation. The performance is evaluated by Root Mean Squared Error (RMSE) to the ground truth concepts for the 1,500 test shots in 961 videos.

We compare our adjustment method with the baseline methods in Table 3.6, where HEX Graph indicates the logical consistent representation [68] on the raw detection scores (i.e. $\beta = 0$), and Group Lasso denotes the representation yield by Eq. (3.1) when $\alpha = 0$. We tune the parameter in each baseline method and report its best performance. As the ground truth label is binary, we let the adjusted scores be binary in all methods. As we see, the proposed method outperforms all baseline methods. We hypothesize the reason is that our method is the only one that combines the distributional consistency and the logical consistency.

We study the parameter sensitivity in the proposed model. Fig. 3.5 plots the RMSE under different parameter settings. Physically, α interpolates the group-wise and within-group sparsity, and β determines the number of concepts in a video. As we see, the parameter β is more sensitive than α, and accordingly we fix the value of α in practice. Note the parameter β is also an important parameter in the baseline methods including thresholding and top-k thresholding.

Table 3.6: Comparison of the adjusted representation and baseline methods on the TRECVID SIN set. The metric is Root Mean Squared Error (RMSE).

Method	RMSE
Raw Score	7.671
HEX Graph Only	8.090
Thresholding	1.349
Top-k Thresholding	1.624
Group Lasso	1.570
Our method	**1.236**



(a) Thresholding (b) Top-k thresholding

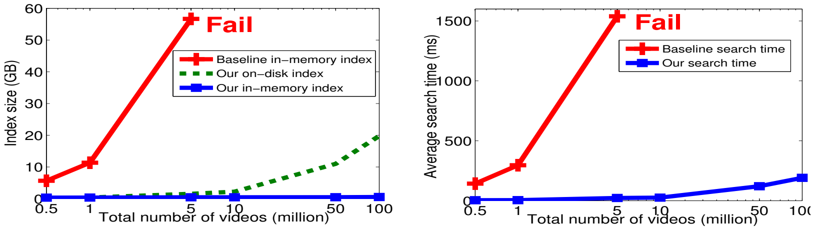
FIGURE 3.5: Sensitivity study on the parameter α and β in our model.

3.6.6 Performance on YFCC100M

We apply the proposed method on YFCC100M, the largest public multimedia collection that has ever been released [59]. It contains about 0.8 million Internet videos (approximately 12 million key shots) on Flickr. For each video and video shot, we extract the improved dense trajectory, and detect 3,000+ concepts by the off-the-shelf detectors in Table 3.1. We implement our inverted index based on Lucene [82], and a similar configuration described in Section 3.6.1 is used except we set $b = 0$ in the BM25 model. All experiments are conducted without using any example or text metadata. It is worth emphasizing that as the dataset is very big. The offline video indexing process costs considerable amount of computational resources in Pittsburgh supercomputing center. To this end, we share this valuable benchmark with our community `http://www.cs.cmu.edu/~lujiang/0Ex/mm15.html`.

To validate the efficiency and scalability, we duplicate the original videos and video shots, and create an artificial set of 100 million videos. We compare the search performance of the proposed method to a common approach in existing studies that indexes the video by dense matrices [55, 58]. The experiments are conducted on a single core of Intel Xeon 2.53GHz CPU with 64GB memory. The performance is evaluated in terms of the memory consumption and the online search efficiency. Fig. 3.6(a) compares the in-memory index as the data size grows, where the x-axis denotes the number of videos in the log scale, and the y-axis measures the index in GB. As we see, the baseline method fails when the data reaches 5 million due to lack of memory. In contrast, our method is scalable and only needs 550MB memory to search 100 million videos. The size of the total inverted index on disk is only 20GB. Fig. 3.6(b) compares the online search speed. We create 5 queries, run each query 100 times, and report the mean runtime in milliseconds. A similar pattern can be observed in Fig. 3.6 that our method is much more efficient than the baseline method and only costs 191ms to process a query on a single core. The above results verify scalability and efficiency of the proposed method.

As a demonstration, we use our system to find relevant videos for commercials. The search is on 800 thousand Internet videos. We download 30 commercials from the Internet, and manually create 30 semantic queries only using semantic visual concepts.



(a) Index (in GB) (b) Search Time (in ms)

Figure 3.6: The scalability and efficiency test on 100 million videos. Baseline method fails when the data reaches 5 million due to the lack of memory. Our method is scalable to 100 million videos.

See detailed results in Table B.3. The ads can be organized in 5 categories. As we see, the performance is much higher than the performance on the MED dataset in Table 3.2. The improvement is a result of the increased data volumes. Fig. 3.7 plots the top 5 retrieved videos are semantically relevant to the products in the ads. The results suggest that our method may be useful in enhancing the relevance of in-video ads.

Table 3.7: Average performance for 30 commercials on the YFCC100M set.

Category	#Ads	Evaluation Metric		
		P@20	MRR	MAP@20
Sports	7	0.88	1.00	0.94
Auto	2	0.85	1.00	0.95
Grocery	8	0.84	0.93	0.88
Traveling	3	0.96	1.00	0.96
Miscellaneous	10	0.65	0.85	0.74
Average	30	0.81	0.93	0.86

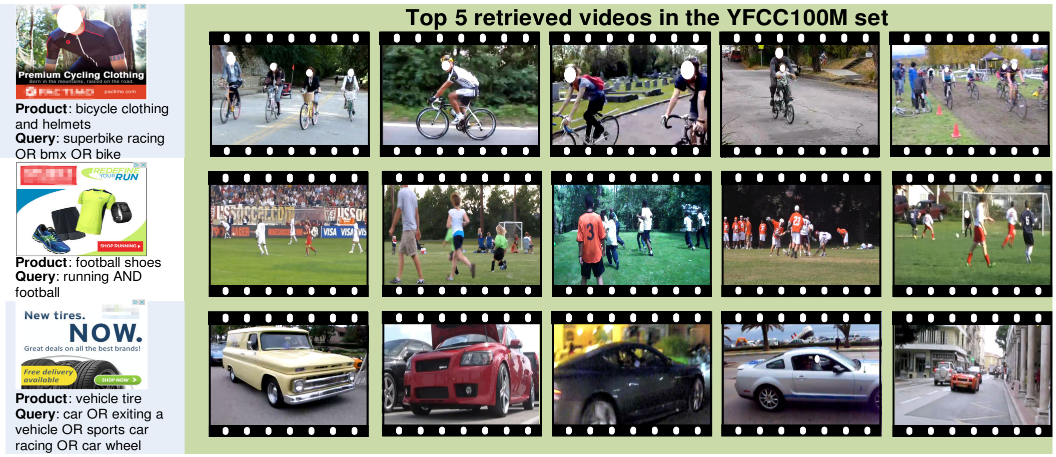

Top 5 retrieved videos in the YFCC100M set
Premium Cycling Clothing
Product: bicycle clothing and helmets
Query: superbike racing OR bmx OR bike
REDEFINE YOUR RUN
Product: football shoes
Query: running AND football
New tires.
NOW.
Great deals on all the best brands!
Free delivery available
Product: vehicle tire
Query: car OR exiting a vehicle OR sports car racing OR car wheel

Figure 3.7: Top 5 retrieved results for 3 example ads on the YFCC100M dataset.

3.7 Summary

This chapter proposed a scalable solution for large-scale semantic search in video. The proposed method extends the current capability of semantic video search by a few orders of magnitude while maintaining state-of-the-art retrieval performance. A key in our solution is a novel step called concept adjustment that aims at representing a video by a few salient and consistent concepts which can be efficiently indexed by the modified inverted index. We introduced a novel adjustment model that is based on a concise optimization framework with solid interpretations. We also discussed a solution that leverages the text-based inverted index for video retrieval. Experimental results validated the efficacy and the efficiency of the proposed method on several datasets. Specifically, the experimental results on the challenging TRECVID MED benchmarks validate the proposed method is of state-of-the-art accuracy. The results on the largest multimedia set YFCC100M set verify the scalability and efficiency over a large collection of 100 million Internet videos.

Chapter 4

Semantic Search

4.1 Introduction

In this chapter, we study the multimodal search process for semantic queries. The process is called semantic search, which is also known as zero-example search [5] or 0Ex for short, as zero examples are provided in the query. Searching by semantic queries is more consistent with human's understanding and reasoning about the task, where an relevant video is characterized by the presence/absence of certain concepts rather than local points/trajectories in the example videos.

We will focus on two subproblems, namely semantic query generation and multimodal search. The semantic query generation is to map the out-of-vocabulary concepts in the user query to their most relevant alternatives in the system vocabulary. The multimodal search component aims at retrieving a ranked list using the multimodal features. We empirically study the methods in the subproblems and share our observations and lessons in building such a state-of-the-art system. The lessons are valuable because of not only the effort in designing and conducting numerous experiments but also the considerable computational resource to make the experiments possible. We believe the shared lessons may significantly save the time and computational cycles for others who are interested in this problem.

4.2 Related Work

A representative content-based retrieval task, initiated by the TRECVID community, is called Multimedia Event Detection (MED) [5]. The task is to detect the occurrence of a main event in a video clip without any textual metadata. The events of interest are mostly daily activities ranging from "birthday party" to "changing a vehicle tire".

The event detection with zero training examples (0Ex) resembles the task of semantic search. 0Ex is an understudied problem, and only few studies have been proposed very recently [11, 53–58, 83]. Dalton et al. [83] discussed a query expansion approach for concept and text retrieval. Habibian et al. [53] proposed to index videos by composite concepts that are trained by combining the labeled data of individual concepts. Wu et al. [55] introduced a multimodal fusion method for semantic concepts and text features. Given a set of tagged videos, Mazloom et al. [54] discussed a retrieval approach to propagate the tags to unlabeled videos for event detection. Singh et al. [84] studied a concept construction method that utilizes pairs of automatically discovered concepts and then prunes those concepts that are unlikely to be helpful for retrieval. Jiang et al. [15, 56] studied pseudo relevance feedback approaches which manage to significantly improve the original retrieval results. Existing related works inspire our system.

4.3 Semantic Search

4.3.1 Semantic Query Generation

Users can express a semantic query in a variety of forms, such as a few concept names, a sentence or a structured description. The Semantic Query Generation (SQG) component translates a user query into a multimodal *system query*, all words of which exist in the *system vocabulary*. A system vocabulary is the union of the dictionaries of all semantic features in the system. The system vocabulary, to some extend, determines what can be detected and thus searched by a system. For ASR/OCR features, the system vocabulary is usually large enough to cover most words in user queries. For semantic visual/audio concepts, however, the vocabulary is usually limited, and addressing the out-of-vocabulary issue is a major challenge for SQG. The mapping between the user and system query is usually achieved with the aid of an ontology such as WordNet and Wikipedia. For example, a user query "golden retriever" may be translated to its most relevant alternative "large-sized dog", as the original concept may not exist in the system vocabulary.

For example, in the MED benchmark, NIST provides a user query in the form of an event-kit description, which includes a name, definition, explication and visual/acoustic evidences. Table 4.1 shows the user query (event kit description) for the event "E011 Making a sandwich". Its corresponding system query (with manual inspection) after SQG is shown in Table 4.2. As we see, SQG is indeed a challenging task as it involves understanding of text descriptions written in natural language.

The first step in SQG is to parse negations in the user query in order to recognize counter-examples. The recognized examples can be either discarded or associated with

Table 4.1: User query (event-kit description) for the event “Making a sandwich”.

Event name		Making a sandwich
Definition		Constructing an edible food item from ingredients, often including one or more slices of bread plus fillings
Explication		Sandwiches are generally made by placing food items on top of a piece of bread, roll or similar item, and placing another piece of bread on top of the food items. Sandwiches with only one slice of bread are less common and are called ”open face sandwiches”. The food items inserted within the slices of bread are known as ”fillings” and often include sliced meat, vegetables (commonly used vegetables include lettuce, tomatoes, onions, bell peppers, bean sprouts, cucumbers, and olives), and sliced or grated cheese. Often, a liquid or semi-liquid ”condiment” or ”spread” such as oil, mayonnaise, mustard, and/or flavored sauce, is drizzled onto the sandwich or spread with a knife on the bread or top of the sandwich fillers. The sandwich or bread used in the sandwich may also be heated in some way by placing it in a toaster, oven, frying pan, countertop grilling machine, microwave or grill. Sandwiches are a popular meal to make at home and are available for purchase in many cafes, convenience stores, and as part of the lunch menu at many restaurants.
Evidences	scene	indoors (kitchen or restaurant or cafeteria) or outdoors (a park or backyard)
	objects/people	bread of various types; fillings (meat, cheese, vegetables), condiments, knives, plates, other utensils
	activities	slicing, toasting bread, spreading condiments on bread, placing fillings on bread, cutting or dishing up fillings
	audio	noises from equipment hitting the work surface; narration of or commentary on the process; noises emanating from equipment (e.g. microwave or griddle)

Table 4.2: System query for the event “E011 Making a sandwich”.

Event	ID	Name	Category	Relevance
Visual	sin346_133	food	man made thing, food	very relevant
	sin346_183	kitchen	structure building, room	very relevant
	yfcc609_505	cooking	human activity, working utensil tool	very relevant
	sin346_261	room	structure building, room	relevant
	sin346_28	bar_pub	structure building,commercial building	relevant
	yfcc609_145	lunch	food, meal	relevant
	yfcc609_92	dinner	food, meal	relevant
ASR	ASR_long	sandwich, food, bread, fill, place, meat, vegetable, cheese, condiment, knife, plate, utensil, slice, toast, spread, cut, dish	-	relevant
OCR	OCR_short	sandwich	-	relevant

a “NOT” operator in the system query. We found that adding counter examples using the Boolean NOT operator tends to improve performance. For example, in the query “Winning a race without a vehicle”, the query including only relevant concepts such as swimming, racing or marathon can achieve an AP of 12.57. However, the query also containing “AND NOT” concepts such as car racing, horse riding or bicycling can achieve an AP of 24.50.

Given an event-kit description, a user query can be represented the event name (1-3 words) or the frequent words in the event-kit description (after removing the template and stop words). This user query can be directly used as the system query for ASR/OCR features as their vocabularies are sufficiently large. For visual/audio concepts, the query are used to map the out-of-vocabulary words to their most relevant concepts in the system vocabulary. To this end, we study the following classical mapping algorithms to map a word in the user query to the concept in the system vocabulary:

Exact word matching: A straightforward mapping is matching the exact query word (usually after stemming) against the concept name or the concept description. Generally, for unambiguous words, this method has high precision but low recall.

WordNet mapping: This mapping calculates the similarity between two words in terms of their distance in the WordNet taxonomy. The distance can be defined in various ways such as structural depths in the hierarchy [85] or shared overlaps between synonymous words [86]. Among the distance metrics, we found the structural depths yields more robust results [85]. WordNet mapping is good at capturing synonyms and subsumption relations between two nouns.

PMI mapping: The mapping calculates the Point-wise Mutual Information (PMI) [87] between two words. Suppose q_i and q_j are two words in a user query, we have:

$$\mathrm{PMI}(q_i; q_j) = \log \frac{P(q_i, q_j | C_{ont})}{P(q_i | C_{ont}) P(q_j | C_{ont})}, \tag{4.1}$$

where $P(q_i|C_{ont}), P(q_j|C_{ont})$ represent the probability of observing q_i and q_j in the ontology C_{ont} (e.g. a collection of Wikipedia articles), which is calculated by the fraction of the document containing the word. $P(q_i, q_j|C_{ont})$ is the probability of observing the document in which q_i and q_j both occur. PMI mapping assumes that similar words tend to co-occur more frequently, and is good at capturing frequently co-occurring concepts (both nouns and verbs).

Word embedding mapping: This mapping learns a word embedding that helps predict the surrounding words in a sentence [88, 89]. The learned embedding, usually by neural network models, is in a lower-dimensional continuous vector space. The cosine coefficient between two words is often used to measure their distance. It is fast and also able to capture the frequent co-occurred words in similar contexts.

We found that discriminating the mapping relevance in a query may increase the performance. In other words, the calculated relevance can be used to weight query terms. For example, the relevance can be categorized into discrete levels according to their relevance to the user query. Table 4.2 have three levels of relevance: “very relevant”, “relevant” and “slightly relevant”, and the levels are assigned to weight of 2.0, 1.0 and

0.5, respectively. We manually modified the relevance produced by the above automatical mapping algorithms. In this way, we can observe an absolute 1-2% improvement over the same query with no weights.

4.3.2 Retrieval Models

Given a system query, the multimodal search component aims at retrieving a ranked list for each modality. We are interested in leveraging the well-studied text retrieval models for video retrieval. This strategy allows us to utilize the infrastructure built for text retrieval. There is no single retrieval model that can work the best for all modalities. As a result, our system incorporates several classical retrieval models and applies them to their most appropriate modalities. Let $Q = q_1, \ldots, q_n$ denote a system query. A retrieval model ranks videos by the score s$(d|Q)$, where d is a video in the video collection C. We study the following retrieval models:

Vector Space Model (VSM): This model represents both a video and a query as a vector of the words in the system vocabulary. The common vector representation includes generic term frequency (tf) and term frequency-inverse document frequency (tf-idf) [90]. s$(d|Q)$ derives from either the dot product or the cosine coefficient between the video and the query vector.

Okapi BM25: This model extends tf-idf representation by:

$$\mathrm{s}(d|Q) = \sum_{i=1}^{n} \log \frac{|C| - df(q_i) + \frac{1}{2}}{df(q_i) + \frac{1}{2}} \frac{tf(q_i, d)(k_1 + 1)}{tf(q_i, d) + k_1(1 - b + b\frac{len(d)}{\overline{len}})}, \tag{4.2}$$

where $|C|$ is the total number of videos in the collection. $df(\cdot)$ calculates the document frequency for a given word in the collection; $tf(q_i, d)$ calculates the raw term frequency for the word q_i in the video d. Unlike text retrieval, in which document frequencies and term frequencies are integers, in multimedia retrieval, these statistics can be real values as concepts are associated with real-valued detection scores. $len(d)$ calculates the sum of concept or word detection scores in the video d, and $\overline{len}$ is the average video length in the collection. k_1 and b are two model parameters to tune [91]. In the experiments, we set $b = 0.75$, and tune k_1 in $[1.2, 2.0]$.

Language Model-JM Smoothing (LM-JM): The score is considered to be generated by a unigram language model [92]:

$$\mathrm{s}(d|Q) = \log P(d|Q) \propto \log P(d) + \sum_{i=1}^{n} \log P(q_i|d), \tag{4.3}$$

where $P(d)$ is usually assumed to be following the uniform distribution, i.e. the same for every video, and can be dropped in the retrieval model. In some cases, we can encode prior information about a video into $P(d)$, such as its view count, length, and viralness [93]. $P(q_i|d)$ is calculated from:

$$P(q_i|d) = \lambda \frac{tf(q_i, d)}{\sum_w tf(w, d)} + (1 - \lambda)P(q_i|C), \tag{4.4}$$

where w enumerates all the words or the concepts in a given video. $P(q_i|C)$ is a smoother that can be calculated by $df(q_i)/|C|$. As we see, Eq. (4.4) linearly interpolates the maximum likelihood estimation (first term) with the collection model (second term) by a coefficient λ. The parameter is usually tuned in the range of $[0.7, 0.9]$. This model is good for retrieving long text queries, e.g. the frequent words in the event kit description.

Language Model-Dirichlet Smoothing (LM-DL): This model adds a conjugate prior (Dirichlet distribution) to the language model:

$$P(q_i|d) = \frac{tf(q_i, d) + \mu P(q_i|C)}{\sum_w tf(w, d) + \mu}, \tag{4.5}$$

where μ is a coefficient balancing the likelihood model and the conjugate prior. It is usually tuned in $[0, 2000]$ [92]. This model is good for short text queries, e.g. the event name.

4.4 Experiments

4.4.1 Setups

Dataset and evaluation: The experiments are conducted on two TRECVID benchmarks called Multimedia Event Detection (MED): MED13Test and MED14Test [5]. The performance is evaluated by the official metric Mean Average Precision (MAP). Each set includes 20 events over 25,000 test videos. The official NIST's test split is used. We also evaluate each experiment on 10 randomly generated splits to reduce the bias brought by the split partition. The mean and 90% confidence interval are reported. All experiments are conducted without using any example or text metadata.

Features and queries: Videos are indexed by semantic features including semantic visual concepts, ASR, and OCR. The same semantic features described in Section 3.6.1 are used in the experiments, except here the features are represented by the raw detection scores before the adjustment. See the details of concept features in Table 3.1. we share our features and experimental results on the benchmark
http://www.cs.cmu.edu/~lujiang/0Ex/icmr15.html.

The user query is the event-kit description. For ASR/OCR, the automatically generated event name and description representations are directly used as the system query. The system query for semantic concepts is obtained by a two-step procedure: a preliminary mapping is automatically generated by the discussed mapping algorithms. The results are then examined by human experts to figure out the final system query. We call these queries *Expert queries*. Besides, we also study the queries automatically generated by the mapping algorithm called "Auto SQG" in Section 4.4.2.

Configurations: In the multimodal search component, by default, the LM-JM model ($\lambda = 0.7$) is used for ASR/OCR for the frequent-words in the event-kit description. BM25 is used for ASR [94] and OCR features for the event name query (1-3 words), where $k_1 = 1.2$ and $b = 0.75$. Both the frequent-words query and the event name query are automatically generated without manual inspection. While parsing the frequent words in the event-kit description, the stop and template words are first removed, and words in the evidence section are counted three times. After parsing, the words with the frequency ≥ 3 are then used in the query. VSM-tf model is applied to all semantic concept features.

In the SQG component, the exact word matching algorithm finds the concept name in the frequent event-kit words (frequency ≥ 3). The WordNet mapping uses the distance metrics in [85] as the default metric. We build an inverted index over the Wikipedia corpus (about 6 million articles), and use it to calculate the PMI mapping. To calculate the statistics, we online issue queries to the index. A pre-trained word embedding trained on Wikipedia [89] is used to calculated the word embedding mapping.

4.4.2 Semantic Matching in SQG

We apply the SQG mapping algorithms in Section 4.3.1 to map the user query to the concepts in the vocabulary. The experiments are conducted only using semantic concept features. We use two metrics to compare these mapping algorithms. One is the precision of the 5 most relevant concepts returned by each algorithm. We manually assess the relevance for 10 events (E031-E040) on 4 concept features (i.e. 200 pairs for each mapping algorithm). The other is the MAP obtained by the 3 most relevant concepts. Table 4.3 lists the results, where the last column lists the runtime of calculating the mapping between 1,000 pairs of words. The second last row (Fusion) indicates the average fusion of the results of all mapping algorithms. As we see, in terms of P@5, PMI is slightly better than others, but it is also the slowest because its calculation involves looking up a index of 6 million text documents in Wikipedia. Fusion of all mapping results yields a better P@5.

We then combine the automatically mapped semantic concepts with the automatically generated ASR and OCR query. Here we assume users have specified which feature to use for each query, and SQG is used to automatically find relevant concepts or words in the specified features. We obtain this information in the event-kit description. For example, we will use ASR when we find words "narration/narrating" and "process" in the event-kit description. An event that has these words in the description tends to be an instructional event, such as "Making a sandwich" and "Repairing an appliance", in which the spoken words are more likely to be detected accurately. Our result can be understood as an overestimate of a fully-automatic SQG system, in which users do not even need to specify the feature. As we see in Table 4.3, PMI performs the best on MED13Test whereas on MED14Test it is Exact Word Matching. The fusion of all mapping results (the second last row) improves the MAP on both the datasets. We then fine-tune the parameters of the mapping fusion and build our AutoSQG system (the last row).

As we see, AutoSQG only achieves about 55% of the full system's MAP. Several reasons account for the performance drop: 1) the concept name does not accurately describe what is being detected; 2) the quality of mapping is limited (P@5=0.42); 3) relevant concepts are not necessarily discriminative concepts. For example, "animal" and "throwing ball" appear to be relevant to the query "playing a fetch", but the former is too general and the latter is about throwing a baseball which is visually different; "dog" is much less discriminative than "group of dogs" for the query "dog show". The results suggest that the automatic SQG is not well-understood. The proposed automatic mappings are still very preliminary, and could be further refined by manual inspection. A significant drawback of current mapping algorithms is representing a concept as a few words. However, in our manual assessment, we regard a concept as a multimodal document that includes a name, description, category, reliability (accuracy) and examples of the top detected video snippet.

Table 4.3: Comparison of SQG mapping algorithms.

Mapping Method	P@5	MAP		Time (s)
		13Test	14Test	
Exact Word Matching	0.340	9.66	7.22	0.10
WordNet	0.330	7.86	6.68	1.22
PMI	0.355	9.84	6.95	22.20
Word Embedding	0.335	8.79	6.21	0.48
Mapping Fusion	0.420	10.22	9.38	-
AutoSQGSys	-	12.00	11.45	-

4.4.3 Modality/Feature Contribution

Table 4.4 compares the modality contribution for semantic search, where each run represents a certain configuration. The MAP is evaluated on MED14Test and MED13Test. As we see, visual modality is the most contributing modality, which by itself can recover about 85% MAP of the full system. ASR and OCR provide complementary contribution to the full system but prove to be much worse than the visual features. The event-level comparison can be found in Table B.4. The experimental results also justify the rationale of multimodal search.

To understand the feature contribution, we conduct leave-one-feature-out experiments. The performance drop, after removing the feature, can be used to estimate its contribution to the full system. As we see in Table 4.5, the results show that every feature provides some contribution. As the feature contribution is mainly dominated by a number of discriminative events, the comparison is more meaningful at the event-level (see Table B.5 and Table B.6), where one can tell that, for example, the contribution of ImageNet mainly comes from three events E031, E015 and E037. Though the MAP drop varies on different events and datasets, the average drop on the two datasets follows: Sports > ImageNet > ASR > IACC > YFCC > DIY > OCR. The biggest contributor Sports is also the most computationally expensive feature to train. In fact, the above order of semantic concepts highly correlates to #samples in their datasets, which suggests the rationale of training concepts over big data sets.

Table 4.4: Comparison of modality contribution for semantic search.

Run	MED13Test		MED14Test	
	1-split	10-splits	1-split	10-splits
FullSys	20.75	19.47±1.19	20.60	18.77±2.16
VisualSys	18.31	18.30±1.11	17.58	17.27±1.82
ASRSys	6.53	6.90±0.74	5.79	4.26±1.19
OCRSys	2.04	4.14±0.07	1.47	2.20±0.73

Table 4.5: Comparison of feature contribution for semantic search.

SysID	Visual Concepts					ASR	OCR	MAP		MAP Drop(%)
	IACC	Sports	YFCC	DIY	ImageNet			1-split	10-splits	
MED13/IACC		✓	✓	✓	✓	✓	✓	18.93	18.61±1.13	9%
MED13/Sports	✓		✓	✓	✓	✓	✓	15.67	14.68±0.92	25%
MED13/YFCC	✓	✓		✓	✓	✓	✓	18.14	18.47±1.21	13%
MED13/DIY	✓	✓	✓		✓	✓	✓	19.95	18.70±1.19	4%
MED13/ImageNet	✓	✓	✓	✓		✓	✓	18.18	16.58±1.18	12%
MED13/ASR	✓	✓	✓	✓	✓		✓	18.48	18.78±1.10	11%
MED13/OCR	✓	✓	✓	✓	✓	✓		20.59	19.12±1.20	1%
MED14/IACC		✓	✓	✓	✓	✓	✓	18.34	17.79±1.95	11%
MED14/Sports	✓		✓	✓	✓	✓	✓	13.93	12.47±1.93	32%
MED14/YFCC	✓	✓		✓	✓	✓	✓	20.05	18.55±2.13	3%
MED14/DIY	✓	✓	✓		✓	✓	✓	20.40	18.42±2.22	1%
MED14/ImageNet	✓	✓	✓	✓		✓	✓	16.37	15.21±1.91	20%
MED14/ASR	✓	✓	✓	✓	✓		✓	18.36	17.62±1.84	11%
MED14/OCR	✓	✓	✓	✓	✓	✓		20.43	18.86±2.20	1%

4.4.4 Comparison of Retrieval Methods

Table 4.6 compares the retrieval models on MED14Test using representative features such as ASR, OCR and two types of visual concepts. As we see, there is no single retrieval model that works the best for all features. For ASR and OCR words, BM25 and Language Model with JM smoothing (LM-JM) yield the best MAPs. An interesting observation is that VSM can only achieve 50% MAP of LM-JM on ASR (2.94 versus

5.79). This observation suggests that the role of retrieval models in semantic search is substantial. For semantic concepts, VSM performs no worse than other models. We hypothesize that it is because the dense raw concept representation, i.e. every dimension has a nonzero value, and this representation is quite different from sparse text features.

To verify this hypothesis, we apply the (top-k) concept adjustment to the Sports feature. We increase the parameter k proportional to the size of vocabulary. As we see, BM25 and LM exhibit better performance in the sparse representations. The results substantiate our hypothesis classical text retrieval algorithms also work for adjusted concept features.

Table 4.6: Comparison of retrieval models on MED14Test using ASR, OCR, Sports and IACC.

Feat.	Split	VSM-tf	VSM-tfidf	BM25	LM-JM	LM-DP
ASR	1	2.94	1.26	3.43	**5.79**	1.45
	10	2.67	1.49	3.03	**4.26**	1.14
OCR	1	0.56	0.47	**1.47**	1.02	1.22
	10	2.50	2.38	**4.52**	3.80	4.07
Sports	1	**9.21**	8.97	8.83	8.75	7.57
	10	**10.61**	10.58	10.13	10.25	9.04
IACC	1	3.49	**3.52**	2.44	2.96	2.06
	10	**2.88**	2.77	2.05	2.45	2.08

Table 4.7: Study of retrieval performance using the adjusted concept features (Sports) on MED14Test.

Density	VSM-tf	BM25	LM-JM	LM-DP
1%	9.06	**9.58**	9.09	9.38
2%	9.93	10.12	**10.14**	10.07
4%	10.34	10.36	10.26	**10.38**
16%	**10.60**	10.45	10.03	9.89
100%	**10.61**	10.13	10.25	9.04

4.5 Summary

In this chapter, we studied semantic search. We focused on two subproblems called semantic query generation and multimodal search. The proposed method goes beyond conventional text-to-text matching, and allows for semantic search without any textual metadata or example videos. We shared our compelling insights on a number of empirical studies. From the experimental results, we found that 1) retrieval models may have substantial impacts to the search result. A reasonable strategy is to incorporate multiple models and apply them to their appropriate features/modalities; 2) automatic query generation for queries in the form of event-kit descriptions is still very challenging. Combining mapping results from various mapping algorithms and applying manual examination afterward is the best strategy known so far.

The methods studied in this chapter is merely a first effort towards semantic search in Internet videos. The proposed method can be improved in various ways, e.g. by incorporating more accurate visual and audio concept detectors, by studying more appropriate retrieval models, by exploring search interfaces or interactive search schemes. As shown in our experiments, the automatic semantic query generation is not well understood. Closing the gap between the manual and automatic query may point to a promising direction.

Chapter 5

Query Embedding and Hybrid Search

5.1 Introduction

This chapter will study two problems: query embedding and hybrid search. Unlike in Chapter 4, both of the problems need training data. In query embedding, we are interested in learning an embedding to help understand user queries for the semantic query generation problem discussed in Chapter 4. In hybrid search, we are interested in finding a method that can accurately process the query with both semantic features and a few video examples.

5.1.1 Query Embedding

Chapter 4 has demonstrated a significant gap between users queries and generic concepts, i.e. a gap between a user's information need and what can be retrieved by the system. This gap becomes a critical issue hindering the delivery of accurate video search. For this problem, we focus on the queries over personal media data, i.e. personal photos and videos. Traditional text-to-text matching approaches, in which query words are matched against images' metadata, are bound to fail on personal media data since about 80% personal media can only be searched via concepts [4]. With the speed that media gets created everyday, manually annotating personal photo archives is practically infeasible. This is a situation comparable to the days in the late 1990s, when people usually got lost in the rising sea of web pages, now they are overwhelmed by the vast amounts of personal media data but lack tools to find desired information.

To bridge this gap, in this chapter we propose novel approaches based on deep query embedding networks that leverages clickthrough data to learn end-to-end mappings directly from personal queries to the automatic concepts. We propose both feed-forward and Recurrent Neural Network (RNN) [95] architectures to examine the effectiveness of sequential modeling. The proposed model implicitly models the complicated nonlinear relations in the visual domain. For example, a user query "birthday party" might not retrieve any results simply because "birthday party" is missing from our concept vocabulary. However, our new approach can translate the query to a set of relevant concepts that exist in the vocabulary, such as "cake", "candle", "kids", etc.

5.1.2 Hybrid Search

The previous chapters have not discussed the method to process the hybrid query, i.e. the query with both semantic features and a few video examples. A straightforward approach to process hybrid queries is to learn two independent models, i.e. a retrieval model discussed in Chapter 4 and a supervised model over a few positive examples, and, later, fuse the model outputs together. This process is called *late fusion*. In our problem, late fusion yields reasonable results but is less accurate when the number of positive samples is small. In this chapter, we study a new method for hybrid query which learns a joint model over both semantic concepts and positive examples. Experimental results show that the proposed method not only outperforms baseline methods, but also proves to be a sparser model which might potentially enable large-scale search over big video data.

5.2 Related Work

Regarding the query embedding, the research of general web search also benefits from the recent progress of modern word embeddings [88]. Very recently, Grbovic et al. [96] used web query embeddings together with advertisement click logs to learn query expansion in a distributed system for query to advertisement matching. In a highly related work, Huang et al. [97] proposed to learn latent semantic models between queries and documents using clickthrough data. We are interested in learning an visual embedding from the queries to automatically generated *concepts*. We employ concepts as the visual information proxy making the learning space significantly smaller. More importantly, we are able to implicitly take into account the *accuracy* of each concept detector; for example, even if a query exists as one of the concepts, the visual detector for that concept might be weak. We may therefore find other concepts to be equally or even more important for ranking than that query concept itself and improve the ranking of results.

We compare against the approach of [97] in Section ??. In some sense, the proposed deep query understanding model might also be the first zero-shot learning approach that uses clickthrough data.

Regarding the hybrid search, to the best of our knowledge, there have been few studies dedicated to study neither hybrid search nor similar types of search, especially in the field of multimedia and computer vision. Generally, hybrid queries are of two parts: semantic features and a few positive examples. A naive way to combine the results is by *late fusion*, i.e. learning different models for different modalities and aggregating the outputs [21].For example, Wu et al. propose a method based on Principle Component Analysis (PCA) and Independent Component Analysis (ICA) to exploit the dependency between different modalities [98]. However, since the method is based on PCA, it assumes 1) that features with large variances are important and 2) that the principal components are orthogonal. As these assumption is too restrictive to hold in many data sets, Rasiwasia el al. [99] proposed to eliminate the dependency by Canonical Correlation Analysis (CCA) i.e. projecting the original feature spaces into a so-called semantic space that maximizes the correlation between different modalities. As CCA only requires a linear relationship between the variable in different modalities, the assumptions of Wu's method can be relaxed. Jiang et al. [21] proposed an approach for high-level and low-level features fusion based on collective classification. Generally, the method consists of three steps: training a classifier from low-level features; encoding high-level features into graphs; and diffusing the scores on the established graph to obtain the final prediction. Another possible approach is to train exemplar SVM classifiers [100] for each positive sample and ensemble the results to obtain the final outputs.

5.3 Query Embedding

5.3.1 Problem

To bridge the gap between user query words and the concepts, we introduce a new method, named Visual Query Embedding (VQE) to improve photo and video search.

Recall, a concept corresponds to a visual recognition model that estimates the probability of observing the concept in the image or video content. There are two major differences between the concepts in images and videos and the words in text documents. First, the concept vocabulary is much smaller than the word vocabulary, limited by the number of objects, scenes or actions that can be accurately detected in the content of photos or videos. Scaling the number of concepts is nontrivial, as training detectors requires considerable amount of labeled data which are expensive to acquire [49]. Second, the accuracy of the automatically detected concepts is limited: the detected concepts may not actually be present whereas concepts not detected may well appear in the content of the photos or videos.

Due to such differences, there is a significant gap between user query words and concepts, i.e. a gap between a user's information need and what can be retrieved by the system. To address this issue, we propose to learn Visual Query Embedding (VQE) models that directly map user query words to the related visual concepts. We propose to address this problem through a novel perspective where end-to-end embeddings are learned leveraging visually relevant concepts discovered in the clickthrough data. Following [97], we assume a query to be relevant, at least partially, to clicked personal media data in that session. Our intuition is that for the same query, concepts frequently occurring in the clicked photos are more likely to be relevant. For example, if many users clicks photos containing the concept "candles" for the query "birthday party", then "candles" is a concept that is probably related to "birthday party". Table 5.1 shows some representative examples discovered form the search logs.

Table 5.1: Examples of user queries and visually relevant concepts.

User queries	Related Visual Concepts
jaguar →	sports car, road
playa →	coast, ocean
bluebell →	flower, purple
tiger →	carnivore, big cat, tiger
andromeda →	empty, dreamlike, fire, bonfire
zoo →	people, animal, primate, dog, monkey

We are interested in learning an end-to-end visual query embedding function from the user query words to the relevant visual concepts discovered in the clickthrough data. Formally, let $Q = q_1, \cdots, q_n$ denote a query of n words, where $Q \in \mathbb{Z}^n$ and each q_i represents an integer index in the query word vocabulary. Define a function $\phi : \mathbb{Z}^n \to \mathbb{R}^m$, where $\mathbb{R}^m$ is a vector over the concept vocabulary of m concepts. Denote $\mathbf{y}_k$ as the relevant concepts extracted from the search log for the query Q_k. Based on the above definitions, we can summarize the visual query embedding problem as a supervised learning problem: $\phi = \arg\min_\phi \sum_k \ell(\phi(Q_k), \mathbf{y}_k)$, where ℓ is the loss function.

In the online search phase, given a user query Q_k, we use ϕ to map it to a vector of relevant concepts, and apply retrieval algorithms to obtain the relevant personal media. In this paper, we employ the vector space (cosine) retrieval model for simplicity, and refer readers to [58] for an analysis on the impact of retrieval algorithms.

5.3.2 Models

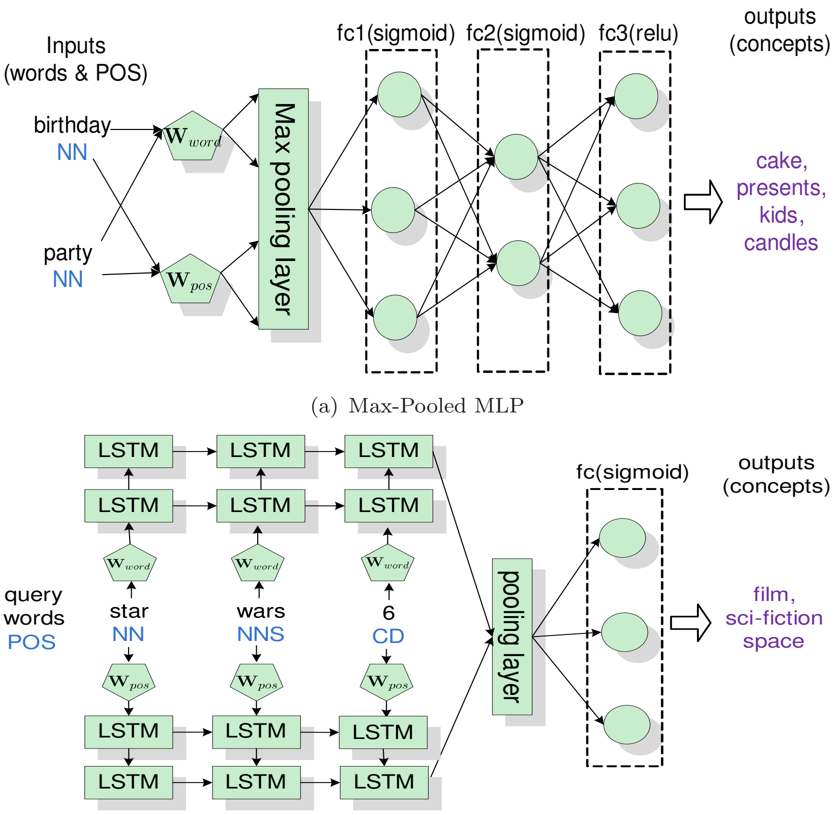
This subsection discusses two deep models for learning the visual query embedding. First of all, we introduce a method to extract visually relevant concepts in a search session. The personal photos or videos are automatically tagged by m concepts in $\mathbb{V}$. Let $\mathbf{d} \in \mathbb{R}^m$ represent the raw detection scores, each dimension of which corresponds to the probability of detecting a concept. For a photo or video, $\mathbf{d}$ is usually a dense vector. In other words, a photo contains almost every concept in the vocabulary with a non-zero detection score. We found learning ϕ based on the dense raw score representation not only leads to worse results but also becomes infeasible for large-scale learning. To address this issue, we incorporate the concept adjustment method in Chapter 3, and represent personal image or video data by the adjusted concept vector $\mathbf{v} \in \mathbb{R}^m$, by:

$$\begin{array}{c} \underset{\mathbf{v}\in[0,1]^m}{\arg\min} \frac{1}{2}\|\mathbf{v}-\mathbf{d}\|_2^2 + \alpha\|\mathbf{v}\|_1 \\ \text{subject to } \mathbf{A}\mathbf{v} \leq \mathbf{0} \end{array}, \tag{5.1}$$

where α is the parameter that controls the sparsity. For simplicity, we use the l_1-norm, and set $\mathbf{A}$ to be the zero matrix as most of the concepts in our experiments are independent.

In the kth session, let C_k^+ represent a set of adjusted concept vectors in the clicked personal media. We define the ground truth vector as the mean of the clicked concept vectors, i.e. $\mathbf{y}_k = 1/|C_k^+| \sum_{\mathbf{v}_i \in C_k^+} \mathbf{v}_i$. Note that the clickthrough data are very noisy [101], containing many queries and clicks made by errors. We found the quality of the training set to greatly affect the accuracy the learned visual query embedding. To reduce noise, we select queries issued by at least 3 users, only consider clicks on the top 30 retrieved results, and the concepts that occur in at least two clicked photos in a session. Within a session, for each concept we compute the mutual information in the set of clicked media and a set of randomly sampled non-clicked media. A concept with lower mutual information means it occurs, indiscriminately, in both clicked and non-clicked sets, and thus is likely to be a background concept such as “outdoor” and “people”. For training, we zero the concepts with small mutual information in the ground-truth vector $\mathbf{y}$.

Given a training set of N sessions, let $\hat{\mathbf{y}}_i$ represent the embedding output after the softmax activation function, i.e. $\hat{\mathbf{y}}_k = \text{softmax}(\phi(Q_k))$, the embedding is learned by minimizing the *cross-entropy loss* function:



(a) Max-Pooled MLP

(b) Two-channel RNN

Figure 5.1: Deep Visual Query Embedding Models.

$$
\begin{aligned}
&\arg\min_{\phi} \sum_{i=1}^{N} \ell(\hat{\mathbf{y}}_i, \mathbf{y}_i) \\
= -&\sum_{i=1}^{N}\sum_{j=1}^{m} \mathbb{1}(y_{ij} > 0)\log \hat{y}_{ij} + \mathbb{1}(y_{ij} = 0)\log(1 - \hat{y}_{ij})
\end{aligned}
\tag{5.2}
$$

where $\mathbb{1}(\cdot)$ is an indicator function equaling 1 when its argument is true, and 0 otherwise. Eq. (5.2) is also known as softmax cross-entropy loss. In the rest of the section, we will discuss two deep neural networks to learn the model.

5.3.2.1 Max-Pooled MLP

Our first model is the max-pooled Multi-Layer Perceptron (MLP), with architecture depicted in Fig. 5.1(a). It takes query words and their Part-of-Speech (POS) tags as input, and outputs the predicted concept vector. The model consists of three types of layers: an embedding layer which maps a word or a POS tag to a low-dimensional vector; a max pooling layer that computes the element-wise maximum for the input vectors; a number of fully connected layer (fc) for nonlinear transformation. Due to the disjoint vocabulary space, we learn separate embeddings $\mathbf{W}_{word}$ for query words and $\mathbf{W}_{pos}$ for POS tags. Denote q_i as the ith word and p_i as its POS tag in the query Q, the model with l layer is calculated from:

$$\begin{aligned} \mathbf{a}_1 &= \max_{q_i, p_i \in Q} (\mathbf{W}_{word}(q_i), \mathbf{W}_{pos}(p_i)) \\ \mathbf{a}_i &= \sigma(\mathbf{W}_i \mathbf{a}_{i-1} + \mathbf{b}_i), \\ \phi(Q) &= \text{relu}(\mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l) \end{aligned} \tag{5.3}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid activation function in the hidden layers, and relu is the rectified linear unit in the last layer. $\mathbf{W}_i$ and $\mathbf{b_i}$ represent the weight matrix and the bias term vector in the ith layer; $\mathbf{a}_i$ is the activation of the ith layer, and $\phi(Q)$ is the predicted concept vector.

5.3.2.2 Two-channel RNN

The word sequence in a query is totally discarded by the max-pooling layer in the previous model. To incorporate the sequence information, we propose a two-channel RNN model. As illustrated in Fig. 5.1(b), the embedding vectors of the word and POS tags are fed into a two layer LSTM units one by one, via two channels: $[q_1, \cdots, q_n, \$]$ and $[p_1, \cdots, p_n, \$]$, where $ is a special token that marks the end of a sequence. LSTM units are used to reduce the vanishing gradients and exploding gradients problem [95]. More precisely, we use the LSTM unit with dropout implementation described in [102]. LSTM updates for time step t, given a word or pos embedding vector as the inputs $\mathbf{x}_t$:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_{xi} x_t + \mathbf{W}_{hi} \mathbf{h}_{t-1} + \mathbf{W}_{c_i} \mathbf{c}_{t-1} + \mathbf{b}_i) \\ \mathbf{f}_t &= \sigma(\mathbf{W}_{xf} \mathbf{x}_t + \mathbf{W}_{hf} \mathbf{h}_{t-1} + \mathbf{W}_{cf} \mathbf{c}_{t-1} + \mathbf{b}_f) \\ \mathbf{c}_t &= \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \tanh(\mathbf{W}_{xc} \mathbf{x}_t + \mathbf{W}_{hc} \mathbf{c}_{h-1} + \mathbf{b}_c) \\ \mathbf{o}_t &= \sigma(\mathbf{W}_{xo} \mathbf{x}_t + \mathbf{W}_{ho} \mathbf{h}_{t-1} + \mathbf{W}_{co} \mathbf{c}_t + \mathbf{b}_o) \\ \mathbf{h}_t &= \mathbf{o}_t \tanh(\mathbf{c}_t) \end{aligned} \tag{5.4}$$

where $\mathbf{i}, \mathbf{f}, \mathbf{o}$ and $\mathbf{c}$ are respectively the input gate, forget gate, output gate and memory cell activation vectors. All of which are the same size of the hidden vector $\mathbf{h}$.

The LSTM hidden states $\mathbf{h}_t$ from the input sequences are fed into an average pooling layer, as shown in Fig. 5.1(b). The pooled hidden states are then fed to a set of fully connected layers similar to those in Eq. (5.3). The final predicted concept vector $\phi(Q)$ is derived from the output of the final fully connected layer.

We incorporate the POS tags in our models for two considerations: first, we found, though adding POS tags would slow down the convergence, in some cases, it helps to find better local minima. The second reason is for generalizability. The proposed models can trivially degenerate to the models without POS tags when tags are less informative.

5.3.3 Results

We conduct our experiments on the Flickr personal search log data. We select personal queries that were issued by at least 3 users, and divide them into a training and a test set according to their issued time. In total, the training set contains about 20,600 personal queries from 3,978 users, while the test set contains 2,443 queries from 1620 users over about 148,000 personal photos. Given a personal query and a photo collection from a user, our goal is to boost the rank for the user clicked photos. We discard all user generated textual metadata that may exist in the user photos in our experiments, and only assume that each photo is tagged with 1,720 automatically detected concepts sampled from the Flickr concept bank [103].

We evaluate performance using two metrics: the non-interpolated mean Average Precision (mAP) of the retrieved ranked list and the concept recall of the top predicted concepts denoted as CR@n. Let $\mathbf{t}$ represents the predicted concepts $\phi(Q)$ after the top-n thresholding, i.e. all elements except for the top n elements in $\phi(Q)$ are set to 0, we have:

$$\text{CR@n}(\mathbf{t}, \mathbf{y}) = \frac{1}{n} \sum_{j=1}^{m} \mathbb{1}(y_i t_i > 0), \tag{5.5}$$

where $\mathbf{y}$ is the ground-truth concept vector extracted in the search session, and $\mathbb{1}$ is the indicator function. Note the two metrics measures different aspects of the search results. mAP evaluates the quality of the clicked photos ranked in the search results, whereas CR@n measures the relevance between the top-n predicted concepts and the true concepts in the clicked photos. A relevant concept may not always lead to a good ranked list as it might be less discriminative, e.g. the relevant concept “carnivore” to the query “tiger”. On the other hand, discriminative concepts leading to better mAP may not always be relevant. Therefore, both metrics are useful in understanding the performance of a method.

Compared Methods: We refer to the two proposed Visual Query Embedding (VQE) models, as VQE (MaxMLP) and VQE (RNN). To demonstrate their performance, we compare them against the following common zero-shot learning and word embedding approaches: **Exact Match** [58] is a plain mapping by matching the exact query words to the concept names. Specifically, it produces a query vector of the same size with the concept vocabulary, each dimension of which represents the similarity between the query and the corresponding concept. The generated query vector is then used to search relevant personal photos. Likewise, **WordNet** computes similarities between query vectors and concepts using WordNet path similarity [104] which is equal to the shortest path in the WordNet taxonomy between the query and the concept name [104]. **SkipGram** [88] learns an embedding space over a large corpus of text documents. In our experiments, the pretrained embedding on Google News is used to compute the query vector. **Semantic DNN** is inspired by the deep semantic structured model of [97], where the authors proposed to learn a low-dimensional embedding space form the query words to the words in the clicked text documents by multi-layer neural networks. In our problem the vocabularies of query and concept are different, and as a result, we add a layer on top of the last layer of the DNN model in [97] to obtain the predicted concept vector. As in [97], the cosine loss function is used to train the model. Note that only the VQE models and the semantic DNN model use the clickthrough training data.

Implementation Details: We implement the proposed VQE models in TensorFlow [105]. The model are trained over mini-batches of 32 samples. The word and POS embeddings are set to 300 dimensional vector and are learned jointly by minimizing the loss in Eq. (5.2). The standard gradient decent algorithm is used to train the MLP models, and the adaptive subgradient (Adagrad) [106] algorithm is used to train the RNN models for faster convergence. The start learning rate is set to 0.1 and is annealed by a staircase exponential decay function with a decay rate of 0.96. A dropout layer is applied in training the RNN networks which discards 0.5% of the input data. Each model is trained at most 7200 epochs (no more than 24 hours).

5.3.3.1 Baseline Comparisons

We first compare the proposed methods with the baseline methods in Table 5.2. As we see, the proposed VQE MaxMLP significantly outperforms other baseline methods. Specifically, it improves the mAP of SkipGram by about relative 45%. We inspected the search results and found that MaxMLP can capture more visually relevant concepts for personal media queries on Flickr. Fig. 5.2 shows representative examples of the top search results for MaxMLP and SkipGram models, where the photos in the green border are the user clicked photos in the search session. As shown in Fig. 5.2(a), MaxMLP retrieves more accurate personal photos. This is because it maps the user query “paint ball” to visually relevant concepts “solider” and “fatigues”, as opposed to the concepts “archery” and “skateboarding” produced by SkipGram. In addition, we found MaxMLP model can find relevant concepts for “who” and “where” quires (see Fig. **??**), the two major categories in personal queries. For example, as shown in Fig. 5.2(c), the MaxMLP model maps the user query “key west”, i.e. a island city, to the concepts “water” and “water sports”, whereas SkipGram fails to find any relevant concept. Besides, experimental results also show the domain difference between learning embedding on clickthrough data versus learning embedding on text corpora like Google News.

Table 5.2: Comparison to baseline methods.

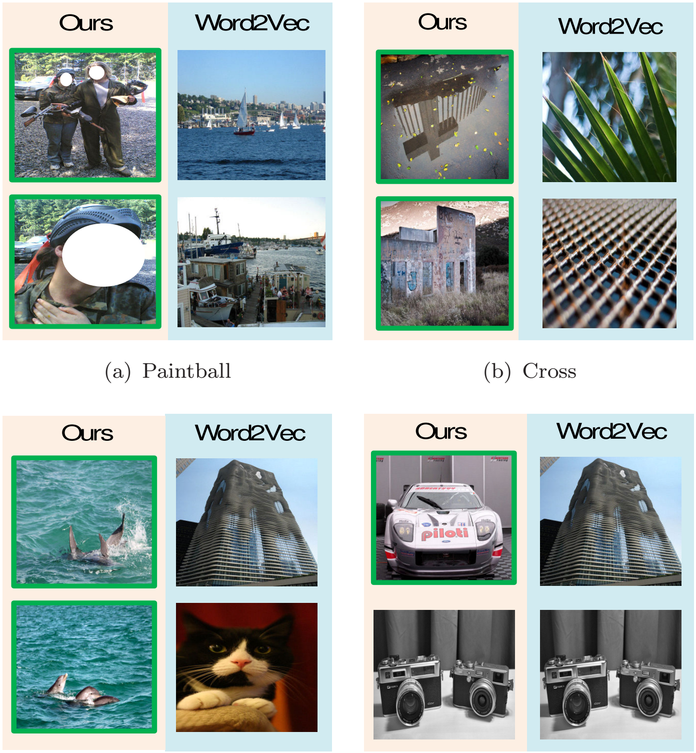
Method	mAP	CR@1	CR@3	CR@5
Exact Match [58]	0.231	0.209	0.086	0.067
WordNet [104]	0.269	0.298	0.195	0.161
SkipGram [88]	0.271	0.286	0.194	0.173
Semantic DNN [97]	0.120	0.010	0.018	0.018
VQE (RNN)	0.235	0.377	0.238	0.167
VQE (MaxMLP)	**0.390**	**0.524**	**0.374**	**0.289**

Although the proposed method shows promising results. We admit that it is still significantly worse than traditional text-to-text search over the photos or videos with rich user-generated metadata. We believe the problem is novel, challenging, and needs further research [58]. We found the lack of common sense often results in inaccurate mappings in the VQE (MaxMLP) model. For example, the user query "bus" is mapped to "tramline" by the VQE model even though there exists a "bus" concept in the vocabulary. This problem may be addressed by either incorporating prior knowledge in training or by increasing the size of the training data. Besides, the worse performance of Semantic DNN model might stem from the less appropriate loss function. See the next subsection for more discussions.

The proposed VQE (RNN) model yields better CR@1 and CR@3 but worse mAP than the baseline SkipGram method, suggesting that the RNN model can find relevant but less discriminative concepts. We found two reasons explaining the worse performance of VQE (RNN) when compared to the VQE (MaxMLP) model: first the worse results suggest the word sequence in personal queries is less informative. It is acknowledged that the sequence of text query words plays a less important role in the bag-of-word or unigram language retrieval model [107]. Our experimental results suggest this may still hold in personal media search. Second, the RNN model converges much slower than the MaxMLP model. When we stopped the training for the RNN model after 24 hours, its performance is still worse than that of the MaxMLP model.

5.3.3.2 Model Parameters

In this section, we study the impact of parameters in the proposed VQE models. First we empirically compare neural network structures. Table 5.3 lists different neural network structures of VQE models, where embedding layers and polling layers are omitted to save space. The detailed model for MaxMLP and RNN model can be found in Eq. (5.3), Eq. (5.4), and Fig. 5.1. For example, the third row MaxMLP4 represents a 4-layer network containing an embedding layer, a pooling layer, a fully connected layer fc_1, transforming a 300d max-polled vector to a hidden layer of 300d by the sigmoid function,



(a) Paintball

(b) Cross

(c) Key west

(d) Ford

Figure 5.2: Top search results of Flickr personal photos. The left ranked list indicates our results and the right list is from the SkipGram (word2vec). The user query is listed in the subtitle, and the photos in the green border are the user clicked photos.

and an output layer fc_2, transforming the 300d hidden vector to an output vector of 1720d by the rectified linear unit. The fifth row MeanMLP4 represents the same network as MaxMLP5 except that it employs the mean instead of the max polling layer .

We observed two trends in Table 5.3. First the performance increases as models get deeper. This observation suggests the visual query embedding for personal media can be highly nonlinear, and deeper models may better capture the underlying relation between user query words and relevant concepts. For example, the 5-layer MaxMLP5 achieves better mAP than the 4-layer MaxMLP4. However, in fact, MaxMLP5 has fewer parameters than MaxMLP4. Second, we found the max polling in the MaxMLP model leads to not only faster convergence but also more accurate search results. For example, MaxMLP5 outperforms MeanMLP5 suggesting the efficacy of the max-polling layer.

The loss function is an important component in neural network training. The softmax cross-entropy loss discussed in Eq. (5.2) represents a type of loss that jointly models concepts as a sparse vector due to the softmax transformation. Alternatively, we can use the cross-entropy loss, which ignores the sparse constraint, or the cosine loss, which measures the distance between queries and concepts seen as dense vectors. Our goal is to find which type of loss is suitable for VQE learning. Table ?? lists the mAP performance. As we see, the cosine loss yields the worst results suggesting treating concepts

TABLE 5.3: Comparison of network structures.

Model	Network Structure	mAP	CR@3
MaxMLP3	fc_1: relu(300 → 1720)	0.225	0.314
MaxMLP4	fc_1: sigmoid(300 → 300) fc_2: relu(300 → 1720)	0.367	0.301
MaxMLP5	fc_1: sigmoid(300 →200) fc_2: sigmoid(200 →200) fc_3: relu(200 →1720)	**0.390**	**0.374**
MeanMLP5	Same as above.	0.249	0.202
RNN3	$lstm_1$ lstm:(300 → 200) fc_1: relu(200 → 1720)	0.124	0.025
RNN6	$lstm_1$ lstm:(300 → 200) $lstm_2$ lstm:(200 → 200) fc_1: sigmoid(200 → 200) fc_2: sigmoid(200 → 200) fc_3: relu(200 → 1720)	0.235	0.238

as dense vectors in the high dimensional space is less appropriate in our problem. This may explain the worse performance of Semantic DNN in Table 5.2. Besides, the comparison between the cross-entropy and the softmax cross-entropy suggests jointly modeling concepts as a sparse representation is helpful.

TABLE 5.4: mAP for different loss functions.

Loss Function	MLP	RNN
Softmax cross-entropy	0.390	0.235
Cross-entropy	0.187	0.145
Cosine distance	0.124	0.130

5.4 Hybrid Search

In this section, we propose a model that learns a joint model of the given semantic features and video examples. See an example of hybrid query in Example 1.1.

5.4.1 Model

Suppose we represent a video by semantic concepts, we can derive the model using the text description without any example videos, or using a few video examples. In other words, the query has two different and independent dual-models.

Because of the dual-models, sometimes, the performance drops even when more training examples are available. See the experiments in Section 5.4.2. Intuitively, there should only be a consistent model for a query. To this end, we propose a method to learn a joint model using both semantic features and video examples.

The key is to find a unified joint representation for the dual-models. Our method is inspired by the Alternating Direction Method of Multipliers (ADMM). Finding a joint model for hybrid query is comparable to learning a parallel machine learning model across a number of machines, where the models independently learned at multiple machines, are aggregated by ADMM into a converged model.

Formally, suppose $\mathbf{X} \in \mathcal{R}^{n \times m}$ is the feature matrix and $\mathbf{y} \in \mathcal{R}^{n}$ is the label vector. Let $\mathbf{w}_0$ denote a linear model, which is a weight vector over the concepts, derived from the text query (zero examples), and $\mathbf{w}_1$ denote the model derived from a few positive video examples. Let L define the loss function. Theoretically, suppose we have an infinite amount of perfect data, the objective function is:

$$\begin{aligned} \text{minimize } & L(\mathbf{X}, \mathbf{y}; \mathbf{w}_1, \mathbf{w}_0) \\ \text{subject to } & \mathbf{w}_0 = \mathbf{w}_1 \end{aligned} \tag{5.6}$$

After introducing an augmented Lagrangian multiplier λ, we have:

$$\begin{aligned} & \text{minimize } L_p(\mathbf{X}, \mathbf{y}; \mathbf{w}_0, \mathbf{w}_1, \lambda) \\ = L(\mathbf{X}, \mathbf{y}; \mathbf{w}_0, \mathbf{w}_1) + \lambda^T(\mathbf{w}_1 - \mathbf{w}_0) + & (p/2)\|\mathbf{w}_1 - \mathbf{w}_0\|^2 \end{aligned} \tag{5.7}$$

We optimize L_p using the ADMM:

$$\begin{aligned} \mathbf{w}_1^{(k+1)} &= \underset{\mathbf{w}_1^{(k)}}{\arg\min}\, L_p(\mathbf{X}, \mathbf{y}; \mathbf{w}_0^{(k)}, \mathbf{w}_1^{(k)}, \lambda^{(k)}) \\ \mathbf{w}_0^{(k+1)} &= \underset{\mathbf{w}_0^{(k)}}{\arg\min}\, L_p(\mathbf{X}, \mathbf{y}; \mathbf{w}_0^{(k)}, \mathbf{w}_1^{(k+1)}, \lambda^{(k)}) \\ \lambda^{(k+1)} &= \lambda^{(k)} + (p/2+1)(\mathbf{w}_1^{(k+1)} - \mathbf{w}_0^{(k+1)}) \end{aligned} \tag{5.8}$$

In this chapter, to simplify the problem, we assume that $\mathbf{w}_0$ is a constant vector given by the user, i.e. $\mathbf{w}_0^{(k)} = \mathbf{w}_0^{(k+1)} = \mathbf{w}_0^{(0)}$. Future work might need to fine-tune $\mathbf{w}_0^{(k)}$. According to [108], Eq. (5.8) is bound to converge. With an infinite amount of perfect data, the optimal solution can be obtained at the convergence. In practice, due to the limited and imperfect data, the optimal solution might be obtained when Eq. (5.8) almost converges. When Eq. (5.8) almost converges, we have $\lambda^{(k)} = \lambda^{(k+1)} \simeq 0$:

$$\begin{aligned} \mathbf{w}_1^{(k+1)} &= \underset{\mathbf{w}_1^{(k)}}{\arg\min}\, L_p(\mathbf{X}, \mathbf{y}; \mathbf{w}_0^{(0)}, \mathbf{w}_1^{(k)}, \lambda^{(k+1)}) \\ &\simeq \underset{\mathbf{w}_1}{\arg\min}\, L(\mathbf{X}, \mathbf{y}; \mathbf{w}_0^{(0)}, \mathbf{w}_1^{(k)}) + p\|\mathbf{w}_1 - \mathbf{w}_0\|^2 \end{aligned} \tag{5.9}$$

In other words, Eq. (5.9) is equivalent to a version of Eq. (5.6) with soft constraints. The hard constraints in Eq. (5.6) is converted to soft constraints by introducing a Lagrangian multiplier p, i.e.:

$$\text{minimize}_{\mathbf{w}_1} L(\mathbf{X}, \mathbf{y}; \mathbf{w}_1, \mathbf{w}_0) + p\|\mathbf{w}_1 - \mathbf{w}_0\|_2^2 \tag{5.10}$$

As a result, we can calculate the final converged model $\mathbf{w}_1^{(k+1)}$ by solving Eq. (5.10). If we use a simple linear-transformation model, we can find the underlying probabilistic model for Eq. (5.10) below. With an infinite amount of perfect data, we have:

$$\begin{aligned} \text{minimize } f(x) &= 1/2\|\mathbf{X}\mathbf{w}_1 - \mathbf{y}\|_2^2 \\ \text{subject to } \mathbf{w}_0 - \mathbf{w}_1 &= 0 \end{aligned} \tag{5.11}$$

However, due to the inaccuracy and incompleteness of the input information, we have

$$\text{minimize } f(x) = 1/2\|\mathbf{X}\mathbf{w}_1 - \mathbf{y}\|_2^2 + p\|\mathbf{w}_0 - \mathbf{w}_1\|_2^2$$

When p is large, the model relies more on the given prior model $\mathbf{w}_0$, and verse versa. The underlying distribution is obvious. Assume $\mathbf{y} = \mathbf{X}\mathbf{w}_1 + \epsilon$, and $\epsilon \sim \mathcal{N}(\mathbf{0}, 2\mathbf{I}^{-1})$ we have:

$$\begin{aligned} P(\mathbf{y}|\mathbf{w_1}) &\propto \exp\{-\frac{1}{2}(\mathbf{X}\mathbf{w_1} - \mathbf{y})^T 2\mathbf{I}(\mathbf{X}\mathbf{w_1} - \mathbf{y})\} \\ &= \exp\{-(\mathbf{X}\mathbf{w_1} - \mathbf{y})^2\} \end{aligned} \tag{5.12}$$

Assume $\mathbf{w}_1$ follows the Gaussian distribution $\mathcal{N}(\mathbf{w}_0, \frac{1}{\sqrt{2p}})$:

$$P(\mathbf{w}_1) \propto \exp\{-p\|\mathbf{w}_1 - \mathbf{w}_0\|_2^2\} \tag{5.13}$$

p represents the inverse of the variance to the known Gaussian mean. Our goal is to maximum a posterior:

$$\begin{aligned} \max_{\mathbf{w_1}} \log P(\mathbf{w_1}|\mathbf{y}) &\propto \log P(\mathbf{y}|\mathbf{w_1}) + \log P(\mathbf{w_1}) \\ \propto \max_{\mathbf{w_1}} \log\exp\{-p\|\mathbf{w}_1 - \mathbf{w}_0\|_2^2\} &+ \log\exp\{-(\mathbf{X}\mathbf{w_1} - \mathbf{y})^2\} \\ &= \min_{\mathbf{w_1}}(\mathbf{X}\mathbf{w_1} - \mathbf{y})^2 + p\|\mathbf{w}_1 - \mathbf{w}_0\|_2^2 \end{aligned} \tag{5.14}$$

As we see, solving Eq. (5.10) is equivalent to maximize a posterior distribution with a conjugated prior distribution.

5.4.2 Experimental Results

Dataset and evaluation: The experiments are conducted on TRECVID benchmark called Multimedia Event Detection (MED): MED14Test [5]. The performance is evaluated by several metrics for a better understanding, which include: P@20, Mean Reciprocal Rank (MRR), mean Average Precision (mAP), and mAP@20, where the mAP is the official metric used by NIST. It includes 20 events over 25,000 test videos. The official NIST's test split is used.

Features and queries: Videos are indexed by semantic features including semantic visual concepts, ASR, and OCR. Three types of semantic features (IACC, YFCC and Sports) described in Section 3.6.1 are used in the experiments. See the details of concept features in Table 3.1. The user query is the event-kit description and 1-10 video examples. The manual query in Chapter 4 is used to obtain the semantic concepts in our experiments.

Baseline Methods: We compare our methods with following baseline methods: **0Ex** represents the baseline model using zero examples, as discussed in Chapter 4. **nEx** represents the best supervised model in [18] trained on n video examples. For example, 1Ex represents the supervised model trained on one example. **Fusion** represents the late fusion of the outputs of 0Ex and nEx. **CCA** follows the idea in [99], where it first projects the outputs of 0Ex and nEx into two latent spaces with the maximum correlation by Canonical Correlation Analysis (CCA) [109] and then it fuses the decorrelated outputs.

Table 5.5 lists the performance of basic individual model using from 0 to 10 video examples. As we see, there is a significant performance drop from 0Ex and 1Ex, i.e. the performance significantly decreases when more examples are available. This counterintuitive observation is because of the disparity of different dual-models learned by semantic features and video examples, in which 0Ex is derived by the method in Chapter 4 whereas 1Ex is trained using only a single example.

TABLE 5.5: Performance of basic individual model of 0Ex and nEx.

	mAP	**P@20**	**MRR**	**mAP@20**
0Ex	0.1278	0.1600	0.4130	0.1099
1Ex	0.0372	0.0675	0.2957	0.0312
2Ex	0.0512	0.0800	0.2753	0.0393
4Ex	0.0856	0.1175	0.3833	0.0628
8Ex	0.1140	0.1600	0.5467	0.0926
10Ex	0.1341	0.1775	0.6139	0.1141

Table 5.6 compares the performance of the baseline and our method using 1 to 10 examples. As we see, our method outperforms all baseline methods across almost all metrics. Especially, when few (1-4) examples are available, our method significantly boosts the mAP. P@20 and MRR. The baseline Fusion method yields reasonable results when there are enough (10+) examples. Figure 5.3 illustrates the mAP in Table 5.6. Note our method is the only one that experiences a continuous growth from 0Ex to 10Ex.

Table 5.6: Comparison of the baseline and our method.

Methods	**mAP**	**P@20**	**MRR**	**mAP@20**
0Ex	0.1278	0.1600	0.4130	0.1099
	1 Examples			
1Ex	0.0372	0.0675	0.2957	0.0312
1Ex+0Ex Fusion	0.0948	0.1325	0.4341	0.0761
1Ex+0Ex CCA	0.0816	0.1100	0.3668	0.0641
1Ex+0Ex Ours	**0.1297**	**0.1700**	**0.5169**	**0.1122**
	2 Examples			
2Ex	0.0512	0.0800	0.2753	0.0393
2Ex+0Ex Fusion	0.1109	0.1350	0.4076	0.0865
2Ex+0Ex CCA	0.1350	0.1600	**0.4977**	0.1110
2Ex+0Ex Ours	**0.1363**	**0.1608**	0.4794	**0.1131**
	4 Examples			
4Ex	0.0856	0.1175	0.3833	0.0628
4Ex+0Ex Fusion	0.1457	0.1650	0.5667	0.1132
4Ex+0Ex CCA	0.1475	0.1800	0.5084	0.1232
4Ex+0Ex Ours	**0.1535**	**0.1900**	**0.5784**	**0.1267**
	8 Examples			
8Ex	0.1140	0.1600	0.5467	0.0926
8Ex+0Ex Fusion	0.1704	0.2150	0.5690	0.1481
8Ex+0Ex CCA	0.1555	0.1950	0.5694	0.1329
Ours	**0.1724**	0**.2220**	**0.6280**	**0.1513**
	10 Examples			
10Ex	0.1341	0.1775	0.6139	0.1141
10Ex+0Ex Fusion	0.1785	**0.2300**	0.6411	0.1588
10Ex+0Ex CCA	0.1595	0.1920	0.5841	0.1366
10Ex+0Ex Ours	**0.1814**	0.2275	0**.6415**	**0.1594**

One benefit of training models on semantic features is interpretability. Because the models are trained on semantic features, we can understand why one model is better than the other. To this end, we inspected three representative events in which our method enjoys the highest mAP gain. We found that the major reason for the improvement is that our method can often find more semantically relevant concepts than the baseline method. Table 5.7 list the highly weighted concepts in the model, where the second and the third column shows the concepts selected by the fusion model and by our model, respectively. As we see, the concepts selected by our model seems more relevant. For example, our model selects the concept “graduation”, “speech”, “talking” and “cheering” for Town Hall Meeting, as opposed to “space”, “concert” and “disgust” selected by the fusion model. Even though the learned concepts largely overlap with the concepts in the user semantic query, our method learns appropriate weights to each relevant concepts. In addition, our model is sparser than the nEx model. For a even sparser model, we can replace the l_2 norm with l_1 norm in Eq. (5.10).

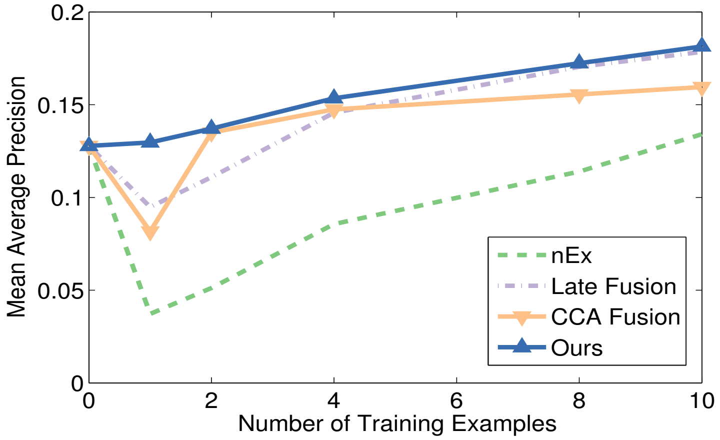


Figure 5.3: Comparison of the baseline and our method.

Table 5.7: Highly weighted concepts selected by the baseline and our model. The concepts are ranked w.r.t their weights. The underlined concepts indicate the concept also appearing in the user semantic query.

ID	Event Name	Late Fusion		Ours	
		mAP	concepts	mAP	concepts
E028	Town hall meeting	0.121	space, concert, disgust, male reporter, reporter	0.228	graduation, speech, talking, cheering, reporter, boredom
E035	Horse riding competition	0.283	equestrianism, endurance riding, kalaripayattu, forest	0.338	show jumping, horse, barrel racing, steeplechase, dressage
E039	Tailgating	0.074	bill, armored vehicle, trip	0.156	tent, truck, van, team, stadium

5.5 Summary

In this chapter, we studied two problems: query embedding and hybrid search. As for query embedding, we proposed novel models for learning visual query embedding between user queries and concepts, and achieved high gains in search performance over existing baselines methods. As for hybrid search, we proposed a method that jointly models trained on semantic using a few examples. Specifically, we employ alternating direction method of multipliers (ADMM) to learn a joint dual-model. The proposed method outperforms existing baseline methods, especially when the training samples are insufficient. Besides, the new model also demonstrates some interpretability over the model trained only on low-level features.

Chapter 6

Multimodal Reranking

6.1 Introduction

Reranking is a technique to improve the quality of search results [110]. The intuition is that the initial ranked result brought by the query has noise which can be refined by the multimodal information residing in the retrieved documents, images or videos. For example, in image search, the reranking is performed based on the results of text-to-text search, in which the initial results are retrieved by matching images' surrounding texts [111]. Studies show that reranking can usually yield improvement of the initial retrieved result [112, 113]. Reranking by multimodal content-based search is still an understudied problem. It is more challenging than reranking by text-to-text search in image search, since the content features not only come from multiple modalities but also are much more noisy. In this chapter, we will introduce two content-based reranking methods, and discuss how they can be united in the same algorithm.

In a generic reranking method, we would first select a few videos, and assign assumed labels to them. Since no ground-truth label is used, the assumed labels are called "*pseudo labels*". The samples with pseudo labels are used to build a reranking model. The statistics collected from the model is used to improve the initial ranked list. Most existing reranking or Pseudo-Relevance Feedback (PRF) methods are designed to construct pseudo labels from a single ranked list, e.g. from the text search [25, 114, 115] or the visual image search [116, 117]. Due to the challenge of multimedia retrieval, features from multiple modalities are usually used to achieve better performance [21, 65]. However, performing multimodal reranking is an important yet unaddressed problem. The key challenge is to jointly derive a pseudo label set from multiple ranked lists. Although reranking may not be a novel idea, reranking by multimodal content-based search is clearly understudied and worthy of exploration, as existing studies mainly concentrate on text-to-text search.

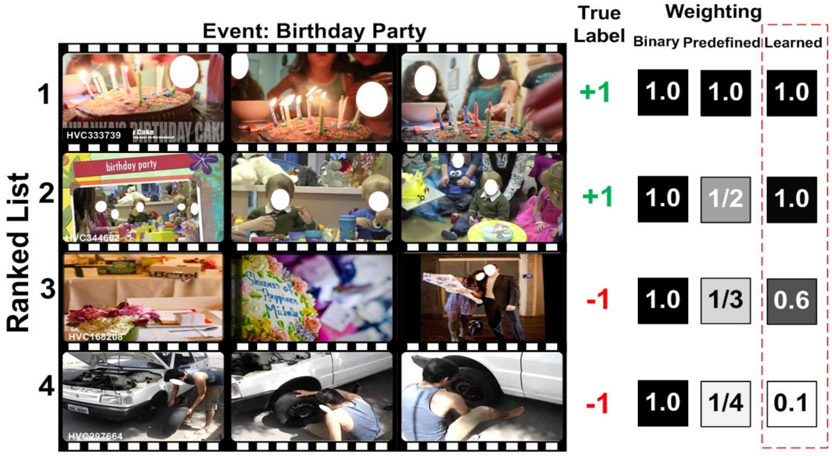


Figure 6.1: Comparison of binary, predefined and learned weights on the query "Birthday Party". All videos are used as positive in reranking. Learned weights are learned by the proposed method.

Besides, an important step in this process is to assign weights to the samples with pseudo labels. The main strategy in current reranking methods is to assign binary (or predefined) weights to videos at different rank positions. These weighting schemes are simple to implement, yet may lead to suboptimal solutions. For example, the reranking methods in [56, 117, 118] assume that top-ranked videos are of equal importance (binary weights). The fact is that, however, videos ranked higher are generally more accurate, and thus more "important", than those ranked lower. The predefined weights [115] may be able to distinguish importance but they are derived independently of reranking models, and thus may not faithfully reflect the latent importance. For example, Fig. 6.1 illustrates a ranked list of videos about "birthday party", where all videos will be used as positive in reranking; the top two are true positive; the third video is a negative but closely related video on wedding shower due to the common concepts such as "gift", "cake" and "cheering"; the fourth video is completely unrelated. As illustrated, neither binary nor predefined weights reflects the latent importance residing in the videos. Another important drawback of binary or predefined weighting is that since the weights are designed based on empirical experience, it is unclear where does, or even whether, the process would converge.

An ideal reranking method would consider the multimodal features and assign appropriate weights in a theoretically sound manner. To this end, we propose two content-based reranking models. The first model is called MultiModal Pseudo Relevance Feedback (MMPRF) which conducts the feedback jointly on multiple modalities leading to a consistent joint reranking model. MMPRF utilizes the ranked lists of all modalities and combines them in a principled approach. MMPRF is a first attempt that leverages both high-level and low-level features for semantic search in a CBVSR system. As we know, it is impossible to use low-level features for semantic search, it is impossible to map the text-like query to the low-level feature without any training data. MMPRF circumvents the difficulty by transferring this problem into a supervised problem on pseudo labels.

The second model is called Self-Paced Reranking (SPaR) which assigns weights adaptively in a self-paced fashion. The method is established on the self-paced learning theory [119, 120]. The theory is inspired by the learning process of humans and animals, where samples are not presented randomly but organized in a meaningful order which illustrates from easy to gradually more complex examples [119]. In the context of reranking problems, easy samples are the top-ranked videos that have smaller loss. As opposed to utilizing all samples to learn a model simultaneously, the proposed model is learned gradually from easy to more complex samples. As the name "self-paced" suggests, in every iteration, SPaR examines the "easiness" of each sample based on what it has already learned, and adaptively determines their weights to be used in the next iteration.

SPaR represents a general multimodal reranking method. MMPRF is a special case of the proposed method that only uses the binary weighting. Compared with existing reranking methods, SPaR has the following three benefits. First, it is established on a solid theory, and of useful properties that can be theoretically verified. For example, SPaR has a concise mathematical objective to optimize, and its convergence property can be theoretically proved. Second, SPaR represents a general framework for reranking on multimodal data, which includes other methods [56, 118, 121], such as MMPRF, as special cases. The connection is useful because once an existing method is modeled as a special case of SPaR, the optimization methods discussed in this chapter become immediately applicable to analyze, and even solve the problem. Third, SPaR offers a compelling insight into reranking by multimodal content-based search [53, 56, 122], where the initial ranked lists are retrieved by content-based search.

The experimental results show promising results on several challenging datasets. As for semantic search, on the MED dataset, MMPRF and SPaR significantly improve the state-of-the-art baseline reranking methods with statistically significant differences; SPaR also outperforms the state-of-the-art reranking methods on an image reranking dataset called Web Query. For hybrid search, SPaR yields statistically significant improvements over the initial search results.

6.2 Related Work

The pseudo labels are usually obtained from a single modality in the literature. On the text modality, reranking, usually known as PRF, has been extensively studied. In the vector space model, the Rocchio algorithm [25] is broadly used, where the original query vector is modified by the vectors of relevant and irrelevant documents. Since a document's true relevance judgment is unavailable, the top-ranked and bottom-ranked documents in the retrieved list are used to approximate the relevant and irrelevant documents. In the language model, PRF is usually performed with a Relevance Model (RM) [114, 123]. The idea is to estimate the probability of a word in the relevance model, and feed the probability back to smooth the query likelihood in the language model. Because the relevance model is unknown, RM assumes the top-ranked documents imply the distribution of the unknown relevance model. Several extensions have been proposed to improve RM. For example, instead of using the top-ranked documents, Lee et al. proposed a cluster-based resampling method to select better feedback documents [124]. Cao et al. explored a supervised approach to select good expansion terms based on a pre-trained classifier [125].

Reranking has also been shown to be effective in image and video retrieval. Yan et al. proposed a classification-based PRF [116–118], where the query image and its most dissimilar images are used as pseudo samples. The idea is to train an imbalanced SVM classifier, biased towards negative pseudo samples, as true negatives are usually much easier to find. In [115], the pseudo negatives, sampled from the ranked list of a text query, are first grouped into several clusters and the clusters' conditional probabilities are fed back to alter the initial ranked list. Similar to [124], the role of clustering is to reduce the noise in the initial text ranked list. In [121, 126], the authors incorporated pseudo labels into the learning to rank paradigm. The idea is to learn a ranking function by optimizing the pair-wise or list-wise orders between pseudo positive and negative samples. In [127], the relevance judgments over the top-ranked videos are provided by users. Then an SVM is trained using visual features represented in the Fisher vector. However, the manual inspection of the search results is prohibited in many problems.

Existing reranking methods are mainly performed based on text-to-text search results, i.e. the initial ranked list is retrieved by text/keyword matching [126, 128]. In terms of the types of the reranking model, these methods can be categorized into Classification, Clustering, Graph and LETOR (LEarning-TO-Rank) based reranking. In Classification-based reranking [118], a classifier is trained upon the pseudo label set, and then tested on retrieved videos to obtain a reranked list. Similarly, in LETOR-based reranking [129] instead of a binary classifier, a ranking function is learned by the pair-wise [121] or list-wise [112, 126] RankSVM. In Clustering-based reranking [115], the retrieved videos are aggregated into clusters, and the clusters' conditional probabilities of the pseudo samples are used to obtain a reranked list. The role of clustering is to reduce the noise in the initial reranking. In Graph-based reranking [130, 131], the graph of retrieved samples needs to be first constructed, on which the initial ranking scores are propagated by methods such as the random walk [21], under the assumption that visually similar videos usually have similar ranks. Generally, reranking methods, including the above methods, are unsupervised methods. There also exist some studies on supervised reranking [111, 128]. Although reranking may not be a novel idea, reranking by multimodal content-based search is clearly understudied and worthy of exploration. Only a few methods have been proposed to conduct reranking based on content-based search results without examples (or training data).

6.3 MMPRF

The intuition behind MMPRF is that the relevant videos can be modeled by a joint discriminative model trained on all modalities. Suppose d_j is a video in the collection, the probability of it being relevant can be calculated from the posterior $P(y_j|d_j;\Theta)$, where y_j is the (pseudo) label for jth video, and Θ denotes the parameter in the joint model. In PRF methods on unimodal data, the partial model is trained on a single modality [116, 117]. We model the ranked list of each modality by its partial model, and our goal is to recover a joint model from these partial models. Formally, we use logistic regression as the discriminative model. For ith modality, the probability of a video being relevant can be calculated from

$$P(y_j|d_j;\Theta_i) = \frac{1}{1+\exp\{-\theta_i^T \mathbf{w}_{ij}\}}, \tag{6.1}$$

where $\mathbf{w}_{ij}$ represents the video d_j's feature vector from the ith modality. $\Theta_i = \theta_i$ denotes the model parameter vector for the ith modality. For a clearer notation, the intercept parameter b is absorbed into the vector θ_i. According to [116], the parameters Θ_i can be independently estimated using the top ranked k^+ samples and the bottom ranked k^- samples in the ith modality, where k^+ and k^- control the number of pseudo positive and pseudo negative samples, respectively.

However, the models estimated independently on each modality can be inconsistent. For example, a video may be used as a pseudo positive in one modality but as a pseudo negative in another modality. An effective approach to find the consistent pseudo label set is by Maximum Likelihood Estimation (MLE) with respect to the label set likelihood over all modalities. Formally, let Ω denotes the union of feedback videos of all modalities. Our objective is to find a pseudo label set that maximizes:

$$\begin{aligned} &\arg\max_{\mathbf{y}} \sum_{i=1}^{m} \ln L(\mathbf{y};\Omega,\Theta_i) \\ &\text{s.t. } ||\mathbf{y}||_1 \le k^+;\ \mathbf{y} \in \{0,1\}^{|\Omega|} \end{aligned} \tag{6.2}$$

where $|\Omega|$ represents the total number of unique pseudo samples, and $\mathbf{y} = [y_1, ...y_{|\Omega|}]^T$ represents their pseudo labels. $L(\mathbf{y}; \Omega, \Theta_i)$ is the likelihood of the label set $\mathbf{y}$ in the ith modality. The sum of likelihood in Eq. (6.2) indicates that each label in the pseudo label set needs to be verified by all modalities and the desired label set satisfies the most modalities. The selection process is analogous to voting, where every modality votes using the likelihood and the better the labels fit a modality, the higher the likelihood is, and vice versa. The set with the highest votes is selected as the pseudo label set. Because each pseudo label is validated by all modalities, the false positives in a single modality may be corrected during the voting. This property is unavailable when only a single modality is considered.

To solve Eq. (6.2), we rewrite the logarithmic likelihood using Eq. (6.1)

$$\begin{aligned} \ln L(\mathbf{y}; \Omega, \Theta_i) &= \ln \prod_{d_j \in \Omega} P(y_j|d_j, \Theta_i)^{y_j} (1 - P(y_j|d_j, \Theta_i))^{(1-y_j)} \\ &= \sum_{j=1}^{|\Omega|} y_j \theta_i^T \mathbf{w}_{ij} - \theta_i^T \mathbf{w}_{ij} - \ln(1 + \exp\{-\theta_i^T \mathbf{w}_{ij}\}) \end{aligned} \tag{6.3}$$

As mentioned above, θ_i can be independently estimated using the top-ranked and bottom-ranked samples in the ith modality. $\mathbf{w}_{ij}$ is the known feature vector. Plugging Eq. (6.3) back to Eq. (6.2) and dropping the constants, the objective function in Eq. (6.3) becomes

$$\begin{aligned} \arg\max_{\mathbf{y}} \sum_{i=1}^{m} \ln L(\mathbf{y}; \Omega, \Theta_i) &= \arg\max_{\mathbf{y}} \sum_{i=1}^{m} \sum_{j=1}^{|\Omega|} y_j \theta_i^T \mathbf{w}_{ij}. \\ &\text{s.t. } ||\mathbf{y}||_1 \leq k^+; \, \mathbf{y} \in \{0,1\}^{|\Omega|} \end{aligned} \tag{6.4}$$

As can be seen, the problem of finding the pseudo label set with the maximum likelihood has been transferred to an integer programming problem, where the objective function is the sum of logarithmic likelihood across all modalities and the pseudo labels are restricted to be binary numbers.

The pseudo negative samples can be randomly sampled from the bottom-ranked samples, as suggested in [115, 116]. In the worst case, suppose n pseudo negative samples are randomly and independently sampled from a collection of samples, and the probability selecting a false negative sample is p. Let the random variable X represents the experiment of selecting pseudo negative samples, then the random variable follows the binomial distribution, i.e. $X \sim B(n, p)$. It is easy to calculate the probability of selecting at least 99% true negatives by

$$F(X \leq 0.01n) = \sum_{i=0}^{\lfloor 0.01n \rfloor} \binom{n}{i} p^i (1-p)^{n-i}, \tag{6.5}$$

where F is the binomial cumulative distribution function. p is usually very small as the number of negative videos is usually far more than that of positive videos. For example, on the MED dataset, $p = 0.003$, and if $n = 100$, the probability of randomly selecting at least 99% true negatives is 0.963. This result suggests that randomly sampled pseudo negatives seems to be sufficiently accurate on the MED dataset.

If the objective function in Eq. (6.2) is calculated from:

$$\ln \hat{L}(\mathbf{y}; \Omega, \Theta_i) = E[\mathbf{y}|\Omega, \Theta_i] = \sum_{j=1}^{|\Omega|} y_j P(y_j|d_j, \Theta_i), \tag{6.6}$$

then the optimization problem in Eq. (6.2) can be solved by the late fusion [132], i.e. the scores in different ranked lists are averaged (or summed) and then the top k^+ videos are selected as pseudo positives. It is easy to verify this yields optimal $\mathbf{y}$ for Eq. (6.2). In fact, late fusion is a common method to combine information within multiple modalities. Eq. (6.2) provides a theoretical justification for the simple method i.e. rather than maximizing the sum of likelihood, one can alternatively maximize the sum of expected values. Note the problem in Eq. (6.2) is tailored to select a small number of accurate labels as opposed to producing a good ranked list in general. Empirically, we observed selecting pseudo positives by the likelihood is better than the expected value when the multiple ranked lists are generated by different retrieval algorithms, e.g. BM25, TFIDF, or Language Model. This is because the distributions of those ranked list (even after normalization) can be quite different. A pain late fusion may produce a biased estimation. In MLE model, estimating Θ in Eq. 6.1 first can put the parameter back into the same scale.

6.4 SPaR

Self-paced Reranking is a general reranking framework for multimedia search. Given a dataset of n samples with features extracted from m modalities, let $\mathbf{x}_{ij}$ denote the feature of the i^{th} sample from the j^{th} modalities, e.g., feature vectors extracted from different channels of a video. $y_i \in \{-1, 1\}$ represents the pseudo label for the i^{th} sample whose values are assumed as the true labels are unknown to reranking methods. The kernel SVM is used to illustrate the algorithm due to its robustness and decent performance in reranking [117]. We will discuss how to generalize it to other models in Section 6.4.3.

Let $\Theta_j = \{\mathbf{w}_j, b_j\}$ denote the classifier parameters for the j^{th} modality, which includes a coefficient vector $\mathbf{w}_j$ and a bias term b_j. Let $\mathbf{v} = [v_1, ..., v_n]^T$ denote the weighting parameters for all samples. Inspired by the self-paced learning [120], suppose n is the total number of samples; m is the total number of modalities; the objective function $\mathbb{E}$ can be formulated as:

$$\begin{aligned}
\min_{\Theta_1,...,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1, ..., \Theta_m, \mathbf{v}, \mathbf{y}; C, k) &= \sum_{j=1}^{m} \min_{\Theta_j,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_j, \mathbf{v}, \mathbf{y}; C, k) \\
= \min_{\substack{\mathbf{y},\mathbf{v},\mathbf{w}_1,...,\mathbf{w}_m,\\ b_1,...,b_m,\{\ell_{ij}\}}} &\sum_{j=1}^{m} \frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n}\sum_{j=1}^{m} v_i\ell_{ij} + mf(\mathbf{v}; k) \\
\text{s.t. } \forall i, \forall j, y_i(\mathbf{w}_j^T \phi(\mathbf{x}_{ij}) + b_j) &\geq 1 - \ell_{ij}, \ell_{ij} \geq 0 \\
\mathbf{y} \in \{-1, +1\}^n&, \mathbf{v} \in [0, 1]^n,
\end{aligned} \tag{6.7}$$

where ℓ_{ij} is the hinge loss, calculated from:

$$\ell_{ij} = \max\{0, 1 - y_i \cdot (\mathbf{w}_j^T \phi(\mathbf{x}_{ij}) + b_j)\}. \tag{6.8}$$

$\phi(\cdot)$ is a feature mapping function to obtain non-linear decision boundaries. C $(C > 0)$ is the standard regularization parameter trading off the hinge loss and the margin. $\sum_{j=1}^{m} v_i\ell_{ij}$ represents the weighted loss for the i^{th} sample. The weight v_i reflects the sample's importance, and when $v_i = 0$, the loss incurred by the i^{th} sample is always zero, i.e. it will not be selected in training.

$f(\mathbf{v}; k)$ is a regularization term that specifies how the samples are selected and how their weights are calculated. It is called the self-paced function as it determines the specific learning scheme. There is an m in front of $f(\mathbf{v}; k)$ as $\sum_{j=1}^{m} f(\mathbf{v}; k) = mf(\mathbf{v}; k)$. $f(\mathbf{v}; k)$ can be defined in various forms which will be discussed in Section 6.4.2. The objective is subjected to two sets of constraints: the first set of constraints in Eq. (6.7) is the soft margin constraint inherited from the conventional SVM. The second constraints in Eq. (6.7) define the domains of pseudo labels and their weights, respectively.

Eq. (6.7) turns out to be difficult to optimize directly due to its non-convexity and complicated constraints. However, it can be effectively optimized by Cyclic Coordinate Method (CCM) [133]. CCM is an iterative method for non-convex optimization, in which the variables are divided into a set of disjoint blocks, in this case two blocks, i.e. classifier parameters $\Theta_1, ..., \Theta_m$, and pseudo labels $\mathbf{y}$ and weights $\mathbf{v}$. In each iteration, a block of variables can be optimized while keeping the other block fixed. Suppose $\mathbb{E}_\Theta$ represents the objective with the fixed block $\Theta_1, ..., \Theta_m$, and $\mathbb{E}_{\mathbf{y},\mathbf{v}}$ represents the objective with the fixed block $\mathbf{y}$ and $\mathbf{v}$. Eq. (6.7) can be solved by the algorithm in Fig. 6.2. In Step 2, the algorithms initializes the starting values for the pseudo labels and weights. Then it optimizes Eq. (6.7) iteratively via Step 4 and 5, until convergence is reached.

```
$t = 0$; //Iteration zero
Choose starting values for $\mathbf{y}, \mathbf{v}$;
while $t \leq$ max iteration do
    $\Theta_1^{(t+1)}, ..., \Theta_m^{(t+1)} = \arg\max \mathbb{E}_{\mathbf{y},\mathbf{v}}(\Theta_1^{(t)}, ..., \Theta_m^{(t)}; C)$;
    $\mathbf{y}^{(t+1)}, \mathbf{v}^{(t+1)} = \arg\max \mathbb{E}_{\Theta}(\mathbf{y}^{(t)}, \mathbf{v}^{(t)}; k)$;
    if $t$ is small then increase $1/k$;
end while
return $[v_1 y_1, \cdots, v_n y_n]^T$;
```

Figure 6.2: Reranking in Optimization Perspective.

```
$t = 0$; //Iteration zero
Choose the initial pseudo labels and weights;
while $t \leq$ max iteration do
    Train a reranking model on the fixed labels and weights;
    Update the pseudo labels and weights;
    if $t$ is small then add more pseudo positives;
end while
return The list of samples after reranking;
```

Figure 6.3: Reranking in Conventional Perspective.

Fig. 6.2 provides a theoretical justification for reranking from the optimization perspective. Fig. 6.3 lists general steps for reranking that have one-to-one correspondence with the steps in Fig. 6.2. The two algorithms present the same methodology from two perspectives. For example, optimizing $\Theta_1, ..., \Theta_m$ can be interpreted as training a reranking model. In the first few iterations, Fig. 6.2 gradually increases the $1/k$ to control the learning pace, which, correspondingly, translates to adding more pseudo positives [56] in training the reranking model.

Fig. 6.2 and Fig. 6.3 offer complementary insights. Fig. 6.2 theoretically justifies Fig. 6.3 on the convergence and the decrease of objective. On the other hand, the empirical experience from studying Fig. 6.3 offers valuable advices on how to set starting values from the initial ranked lists, which is less concerned in the optimization perspective. According to Fig. 6.3, to use SPaR one needs to alternate between two steps: training reranking models and determining the pseudo samples and their weights for the next iteration. We will discuss how to optimize $\mathbb{E}_{\mathbf{y},\mathbf{v}}$ (training reranking models on pseudo samples) in Section 6.4.1, and how to optimize $\mathbb{E}_{\Theta}$ (selecting pseudo samples and their weights based on the current reranking model) in Section 6.4.2.

6.4.1 Learning with Fixed Pseudo Labels and Weights

With the fixed $\mathbf{y}, \mathbf{v}$, Eq. (6.7) represents the sum of weighted hinge loss across all modalities, i.e,

$$\begin{aligned} &\min_{\Theta_1,...,\Theta_m} \mathbb{E}_{\mathbf{y},\mathbf{v}}(\Theta_1, ..., \Theta_m; C) \\ = &\min_{\mathbf{w}_1,...,\mathbf{w}_m, b_1,...,b_m, \{\ell_{ij}\}} \sum_{j=1}^{m} \frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n}\sum_{j=1}^{m} v_i \ell_{ij} \\ &\text{s.t. } \forall i, \forall j, y_i(\mathbf{w}_j^T \phi(\mathbf{x}_{ij}) + b_j) \geq 1 - \ell_{ij}, \ell_{ij} \geq 0. \end{aligned} \tag{6.9}$$

As mentioned, $v_i \ell_{ij}$ is the discounted hinge loss of the i^{th} sample from the j^{th} modality. Eq. (6.9) represents a non-conventional SVM as each sample is associated with a weight reflecting its importance. Eq. (6.9) is non-trivial to optimize directly due to its complex constraints. As a result, we introduce a method that finds the optimum solution for Eq. (6.9). The objective of Eq. (6.9) can be decoupled, and each modality can be optimized independently. Now consider the j^{th} modality ($j = 1, ..., m$). We introduce Lagrange multipliers λ and α, and define the Lagrangian of the problem as:

$$\begin{aligned} \Lambda(\mathbf{w}_j, b_j, \alpha, \lambda) &= \frac{1}{2}\|\mathbf{w}_j\|_2^2 + C\sum_{i=1}^{n} v_i \ell_{ij} \\ + \sum_{i=1}^{n} \alpha_{ij}(1 - \ell_{ij} - y_i \mathbf{w}_j^T \phi(\mathbf{x}_{ij}) - y_i b_j) &+ \sum_{i=1}^{n} -\lambda_{ij}\ell_{ij} \\ &\text{s.t. } \forall i, \alpha_{ij} \geq 0, \lambda_{ij} \geq 0. \end{aligned} \tag{6.10}$$

Since only the j^{th} modality is considered, j is a fixed constant. The Slater's condition trivially holds for the Lagrangian, and thus the duality gap vanishes at the optimal solution. According to the KKT conditions [134], the following conditions must hold for its optimal solution:

$$\begin{aligned} \frac{\nabla \Lambda}{\mathbf{w}_j} = \mathbf{w}_j - \sum_{i=1}^{n} \alpha_{ij} y_i \phi(\mathbf{x}_{ij}) = \mathbf{0}, \frac{\nabla \Lambda}{b_j} &= \sum_{i=1}^{n} \alpha_{ij} y_i = 0, \\ \forall i, \frac{\partial \Lambda}{\partial \ell_{ij}} &= C v_i - \alpha_{ij} - \lambda_{ij} = 0. \end{aligned} \tag{6.11}$$

According to Eq. (6.11), $\forall i, \lambda_{ij} = Cv_i - \alpha_{ij}$, and since Lagrange multipliers are nonnegative, we have $0 \leq \alpha_{ij} \leq Cv_i$. Substitute these inequations and Eq. (6.11) back into Eq. (6.10), the problem's dual form can be obtained by:

$$\begin{aligned} \max_{\alpha} \sum_{i=1}^{n} \alpha_{ij} - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n} \alpha_{ij}\alpha_{kj} y_i y_k \kappa(\mathbf{x}_{ij}, \mathbf{x}_{kj}), \\ \text{s.t. } \sum_{i=1}^{n} y_i \alpha_{ij} = 0, 0 \leq \alpha_{ij} \leq C v_i, \end{aligned} \tag{6.12}$$

where $\kappa(\mathbf{x}_{ij}, \mathbf{x}_{kj}) = \phi(\mathbf{x}_{ij})^T \phi(\mathbf{x}_{kj})$ is the kernel function. Compared with the dual form of a conventional SVM, Eq. (6.12) imposes a sample-specific upper-bound on the support vector coefficient. A sample's upper-bound is proportional to its weight, and therefore a sample with a smaller weight v_i is less influential as its support vector coefficient is bounded by a small value of Cv_i. Eq. (6.12) degenerates to the dual form of conventional SVMs when $\mathbf{v} = \mathbf{1}$. According to the Slater's condition, strong duality holds, and therefore Eq. (6.10) and Eq. (6.12) are equivalent problems. Since Eq. (6.12) is a quadratic programming problem in its dual form, there exists a plethora of algorithms to solve it [134].

6.4.2 Learning with Fixed Classification Parameters

With the fixed classification parameters $\Theta_1, ..., \Theta_m$, Eq. (6.7) becomes:

$$\min_{\mathbf{y},\mathbf{v}} \mathbb{E}_{\Theta}(\mathbf{y}, \mathbf{v}; k) = \min_{\mathbf{y},\mathbf{v}} C \sum_{i=1}^{n} \sum_{j=1}^{m} v_i \ell_{ij} + m f(\mathbf{v}; k) \qquad \text{s.t. } \mathbf{y} \in \{-1, +1\}^n, \mathbf{v} \in [0,1]^n. \tag{6.13}$$

The goal of Eq. (6.13) is to learn not only the pseudo labels $\mathbf{y}$ but also their weights $\mathbf{v}$. Note, as discussed in Section 6.3, the pseudo negative samples can be randomly sampled. In this section, the learning process focuses on pseudo positive samples. Learning $\mathbf{y}$ is easier as its optimal values are independent of $\mathbf{v}$. We first optimize each pseudo label by:

$$y_i^* = \mathop{\arg\min}_{y_i=\{+1,-1\}} \mathbb{E}_{\Theta}(\mathbf{y}, \mathbf{v}) = \mathop{\arg\min}_{y_i=\{+1,-1\}} C \sum_{j=1}^{m} \ell_{ij}, \tag{6.14}$$

where y_i^* denotes the optimum for the i^{th} pseudo label. Solving Eq. (6.14) is simple as all labels are independent with each others in the sum, and each label can only take binary values. Its global optimum can be efficiently obtained by enumerating each y_i. For n samples, we only need to enumerate $2n$ times. In practice, we may need to tune the model to ensure there are a number of pseudo positives.

Having found the optimal $\mathbf{y}$, the task switches to optimizing $\mathbf{v}$. Recall $f(\mathbf{v}; k)$ is the self-paced function, and in [120], it is defined as the l_1 norm of $\mathbf{v} \in [0,1]^n$.

$$f(\mathbf{v}; k) = -\frac{1}{k} \|\mathbf{v}\|_1 = -\frac{1}{k} \sum_{i=1}^{n} v_i. \tag{6.15}$$

Substituting Eq. (6.15) back into Eq. (6.13), the optimal $\mathbf{v}^* = [v_1^*, ..., v_n^*]^T$ is then calculated from

$$v_i^* = \begin{cases} 1 & \frac{1}{m} \sum_{j=1}^{m} C \ell_{ij} < \frac{1}{k} \\ 0 & \frac{1}{m} \sum_{j=1}^{m} C \ell_{ij} \geq \frac{1}{k}. \end{cases} \tag{6.16}$$

The underlying intuition of the self-paced learning can be justified by the closed-form solution in Eq. (6.16). If a sample's average loss is less than a certain threshold, $1/k$ in this case, it will be selected, or otherwise not be selected, as a training example. The

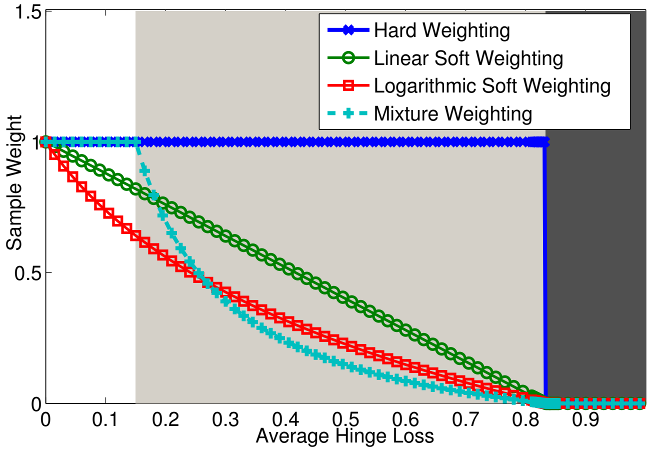


FIGURE 6.4: Comparison of different weighting schemes ($k = 1.2$, $k' = 6.7$). Hard Weighting assigns binary weights. The figure is divided into 3 colored regions, i.e. "white", "gray" and "black" in terms of the loss.

parameter k controls the number of samples to be included in training. Physically, $1/k$ corresponds to the "age" of the model. When $1/k$ is small, only easy samples with small loss will be considered. As $1/k$ grows, more samples with larger loss will be gradually appended to train a "mature" reranking model.

As we see in Eq. (6.16), the variable $\mathbf{v}$ takes only binary values. This learning scheme yields a hard weighting as a sample can be either selected ($v_i = 1$) or unselected ($v_i = 0$). The hard weighting is less appropriate in our problem as it cannot discriminate the importance of samples, as shown in Fig. 6.4. Correspondingly, the soft weighting, which assigns real-valued weights, reflects the latent importance of samples in training more faithfully. The comparison is analogous to the hard/soft assignment in Bag-of-Words quantization, where an interest point can be assigned either to its closest cluster (hard), or to a number of clusters in its vicinity (soft). We discuss three of them, namely, linear, logarithmic and mixture weighting. Note that the proposed functions may not be optimal as there is no single weighting scheme that can always work the best for all datasets.

Linear soft weighting: Probably the most common approach is to linearly weight samples with respect to their loss. This weighting can be realized by the following self-paced function:

$$f(\mathbf{v}; k) = \frac{1}{k}(\frac{1}{2}\|\mathbf{v}\|_2^2 - \sum_{i=1}^{n} v_i). \tag{6.17}$$

Considering $v_i \in [0, 1]$, the close-formed optimal solution for v_i ($i = 1, 2, ..., n$) can be written as:

$$v_i^* = \begin{cases} -k(\frac{1}{m}\sum_{j=1}^{m} C\ell_{ij}) + 1 & \frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} < \frac{1}{k} \\ 0 & \frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} \geq \frac{1}{k}. \end{cases} \tag{6.18}$$

Similar as the hard weighting in Eq. (6.16), the weight is 0 for the samples whose average loss is larger than $1/k$; Otherwise, the weight is linear to the loss (see Fig. 6.4).

Logarithmic soft weighting: The linear soft weighting penalizes the weight linearly in terms of the loss. A more conservative approach is to penalize the weight logarithmically, which can be achieved by the following function:

$$f(\mathbf{v};k) = \sum_{i=1}^{n}(\zeta v_i - \frac{\zeta^{v_i}}{\log \zeta}), \tag{6.19}$$

where $\zeta = (k-1)/k$ and $k > 1$. The closed-form optimal is then given by:

$$v_i^* = \begin{cases} \frac{1}{\log \zeta}\log(\frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} + \zeta) & \frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} < \frac{1}{k} \\ 0 & \frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} \geq \frac{1}{k}. \end{cases} \tag{6.20}$$

Mixture weighting: Mixture weighting is a hybrid of the soft and the hard weighting. One can imagine that the loss range is divided into three colored areas, as illustrated in Fig. 6.4. If the loss is either too small ("white" area) or too large ("black" area), the hard weighting is applied. Otherwise, for the loss in the "gray" area, the soft weighting is applied. Compared with the soft weighting scheme, the mixture weighting tolerates small errors up to a certain point. To define the start of the "gray" area, an additional parameter k' is introduced. Formally,

$$f(\mathbf{v};k,k') = -\zeta \sum_{i=1}^{n} \log(v_i + \zeta k), \tag{6.21}$$

where $\zeta = \frac{1}{k'-k}$ and $k' > k > 0$. The closed-form optimal solution is given by:

$$v_i^* = \begin{cases} 1 & \frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} \leq \frac{1}{k'} \\ 0 & \frac{1}{m}\sum_{j=1}^{m} C\ell_{ij} \geq \frac{1}{k} \\ \frac{m\zeta}{\sum_{j=1}^{m} C\ell_{ij}} - k\zeta & \text{otherwise.} \end{cases} \tag{6.22}$$

Eq. (6.22) tolerates any loss lower than $1/k'$ by assigning the full weight. It penalizes the weight by the inverse of the loss for samples in the "gray" area which starts from $1/k'$ and ends at $1/k$ (see Fig. 6.4). The mixture weighting has the properties of both hard and soft weighting schemes. The comparison of these weighting schemes is listed in the toy example below.

Example 6.1. *Suppose we are given six samples from two modalities. The hinge loss of each sample calculated by Eq. (6.8) is listed in the following table, where Loss*$_1$ *and Loss*$_2$ *column list the losses w.r.t. the first and the second modality, whereas "Avg Loss"* *column lists the average loss. The last four columns present the weights calculated by Eq. (6.16), Eq. (6.18), Eq. (6.20) and Eq. (6.22) where $k = 1.2$ and $k' = 6.7$.*

ID	*Loss$_1$*	*Loss$_2$*	*Avg Loss*	*Hard*	*Linear*	*Log*	*Mixture*
1	*0.08*	*0.02*	*0.05*	*1*	*0.940*	*0.853*	*1.000*
2	*0.15*	*0.09*	*0.12*	*1*	*0.856*	*0.697*	*1.000*
3	*0.50*	*0.50*	*0.50*	*1*	*0.400*	*0.226*	*0.146*
4	*0.96*	*0.70*	*0.83*	*1*	*0.004*	*0.002*	*0.001*
5	*0.66*	*1.02*	*0.84*	*0*	*0.000*	*0.000*	*0.000*
6	*1.30*	*1.10*	*1.20*	*0*	*0.000*	*0.000*	*0.000*

As we see, Hard produces less reasonable solutions, e.g. the difference between the first (ID=1) and the fourth sample (ID=4) is 0.78 and they share the same weight 1; on the contrary, the difference between the fourth and the fifth sample is only 0.01, but suddenly they have totally different weights. This abrupt change is absent in other weighting schemes. Log is a more prudent scheme than Linear as it diminishes the weight more rapidly. Among all weighting schemes, Mixture is the only one that tolerates small errors.

6.4.3 Convergence and Relation to Other Reranking Models

The proposed SPaR has some useful properties. The following lemma proves that the optimum solution can be obtained for the proposed self-paced functions.

Lemma 6.1. *For the self-paced functions in Section 6.4.2, the proposed method finds the optimal solution for Eq. (6.13).*

The following theorem proves the convergence of algorithm 6.2.

Theorem 6.2. *The algorithm in Fig. 6.2 converges to a stationary solution for any fixed C and k.*

A general form of Eq. (6.7) is written as

$$\begin{aligned} \min_{\Theta_1,...,\Theta_m,\mathbf{y},\mathbf{v}} \mathbb{E}(\Theta_1,...,\Theta_m,\mathbf{v},\mathbf{y};k) = \\ \min_{\Theta_1,...,\Theta_m,\mathbf{y},\mathbf{v}} \sum_{i=1}^{n}\sum_{j=1}^{m} v_i Loss(\mathbf{x}_{ij};\Theta_j) + mf(\mathbf{v};k) \\ \text{s.t. Constraints on } \Theta_1,...,\Theta_m, \mathbf{y} \in \{-1,+1\}^n, \mathbf{v} \in [0,1]^n, \end{aligned} \tag{6.23}$$

where $Loss(\mathbf{x}_{ij};\Theta_j)$ is a general function of the loss incurred by the i^{th} sample against the j^{th} modality, e.g., it is defined as the sum of the hinge loss and the margin in

Eq. (6.7). The constraints on $\Theta_1, ..., \Theta_m$ are the constants in the specific reranking model. Alg. 6.2 is still applicable to solve Eq. (6.23). In theory, Eq. (6.23) can be used to find both pseudo positive, pseudo negative samples, and their weights. In practice, we recommend only learning pseudo positive samples and their weights by Eq. (6.23).

Eq. (6.23) represents a general reranking framework, which includes existing reranking methods as special cases. For example, generally, when $Loss$ takes the negative likelihood of Logistic Regression, and $f(\mathbf{v}; k)$ takes Eq. (6.15) (hard weighting scheme), SPaR corresponds to MMPRF. When $Loss$ is the hinge loss, $f(\mathbf{v}; k)$ is Eq. (6.15), the pseudo labels are assumed to be +1, and there is only one modality, SPaR corresponds to Classification-based PRF [117, 118]. Given $Loss$ and constraints on Θ are from pair-wise RankSVM, SPaR can degenerate to LETOR-based reranking methods [121].

6.5 Experiments

6.5.1 Setups

Dataset, query and evaluation: We conduct experiments on the TRECVID Multimedia Event Detection (MED) set including around 34,000 videos on 20 Pre-Specified events. The used queries are semantic queries discussed in the previous chapters. The performance is evaluated on the MED13Test consisting of about 25,000 videos, by the official metric Mean Average Precision (MAP). The official test split released by NIST is used. No ground-truth labeled videos are used in all experiments. In the baseline comparison, we evaluate each experiment 10 times on randomly generated splits to reduce the bias brought by the partition. The mean and 90% confidence interval are reported.

Features: The used semantic features include Automatic Speech Recognition (ASR), Optical Character Recognition (OCR), Semantic INdexing (SIN) and DCNN (Deep Convolutional Neural Network). SIN and DCNN [47] include 346 visual concepts and 1,000 visual objects trained on TRECVID and ImageNet sets. Two types of low-level features are used: dense trajectories [46] and MFCCs.

Baselines: The proposed method is compared against the following baselines: 1) *Without Reranking* is a plain retrieval method without Reranking, and the language model with Jelinek-Mercer smoothing is used [107]. 2)*Rocchio* is a classical reranking model for vector space model under tf-idf representation [25]. 3) *Relevance Model* is a well-known reranking method for text, and the variant with the i.i.d. assumption in [114] is used. 4)*CPRF*(Classification-based PRF) is a seminal PRF-based reranking method. Following [117, 118], SVM classifiers with the χ^2 kernel are trained using the top-ranked and bottom-ranked videos [118]. 5)*Learning to Rank* is a LETOR-based method. Following [121], it is trained using the pairwise constraints derived from the pseudo-positives and pseudo-negatives. A LambdaMART [135] in the RankLib toolkit is used to train the RankSVM model; The parameters of all methods, including the proposed SPaR, are tuned on a third dataset that shares no overlap with our development set.

Model Configuration: Alg. 6.2 is used to solve MMPRF and SPaR. In MMPRF, lp_solve [136] is used to solve the linear/integer programming problem. The regression with the elastic net regularization [137] is used to estimate the parameters of the partial models. Linear and χ^2 kernel is used for dense trajectory and MFCCs features. By default, 10 pseudo positive samples are selected by Eq. (6.2) in MMPRF MLE model. A hundred of pseudo-negatives were randomly sampled from the bottom of the fused ranked list. For a fair comparison, we fix the pseudo negative samples used in all baseline methods.

In SPaR, Eq. (6.12) is solved by the quadratic programming package "quadprog" [138], in which the parameter C is fixed to 1 and the ϕ is set as the χ^2 explicit feature map [139]. By default, Eq. (6.21) is used. The initial values of the pseudo positive labels and weights are derived by MMPRF. Since, according to [56], pseudo negative samples have little impact on the MAP, Eq. (6.12) is only used to learn pseudo positive samples.

6.5.2 Comparison with Baseline methods

We first examine the overall MAP in Table 6.1, in which the best result is highlighted. As we see, MMPRF significantly outperforms the baseline method without PRF. SPaR outperforms all baseline methods by statistically significant differences. For example, on the NIST's split, it increases the MAP of the baseline without reranking by a relative 230% (absolute 9%), and the second best method MMPRF by a relative 28% (absolute 2.8%). Fig. 6.6 plots the AP comparison on each event, where the x-axis represents the event ID and the y-axis denotes the average precision. As we see, SPaR outperforms the baseline without reranking on 18 out of 20 events, and the second best MMPRF on 15 out of 20 events. The improvement is statistically significant at the p-level of 0.05, according to the paired t-test. Fig. 6.5 illustrates the top retrieved results on two events that have the highest improvement. As we see, the videos retrieved by SPaR are more accurate and visually coherent.

We observed two reasons accounting for the improvement brought by MMPRF. First, MMPRF explicitly considers multiple modalities and thus can produce a more accurate pseudo label set. Second, the performance of MMPRF is further improved by leveraging both high-level and low-level features. The improvement of SPaR stems from the capability of adjusting weights of pseudo samples in a reasonable way. For example,

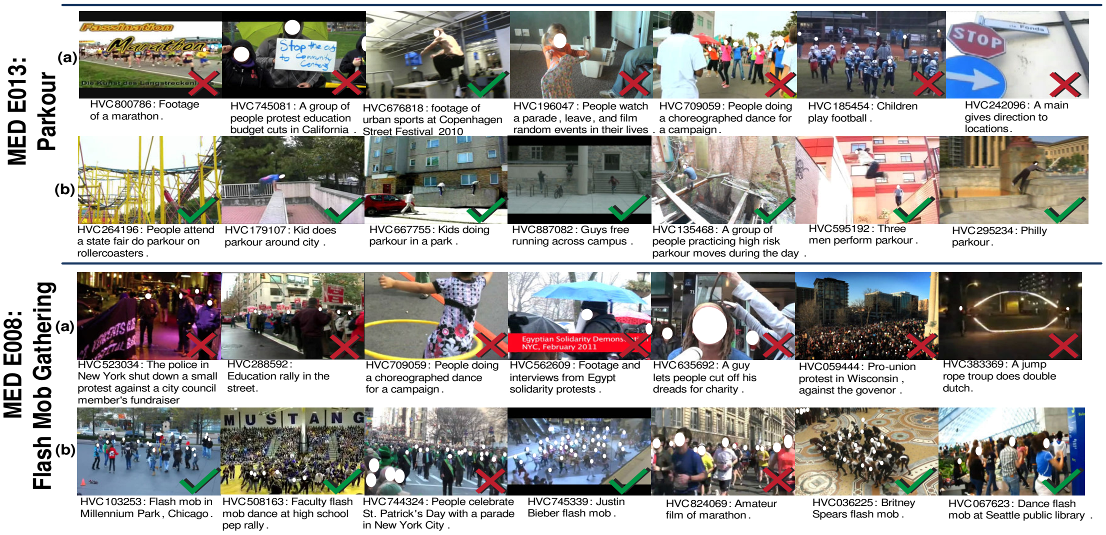

MED E013: Parkour
(a)
HVC800786: Footage of a marathon.
HVC745081: A group of people protest education budget cuts in California .
HVC676818: footage of urban sports at Copenhagen Street Festival 2010
HVC196047: People watch a parade, leave, and film random events in their lives .
HVC709059: People doing a choreographed dance for a campaign.
HVC185454: Children play football.
HVC242096: A main gives direction to locations.
(b)
HVC264196: People attend a state fair do parkour on rollercoasters.
HVC179107: Kid does parkour around city .
HVC667755: Kids doing parkour in a park .
HVC887082: Guys free running across campus .
HVC135468: A group of people practicing high risk parkour moves during the day .
HVC595192: Three men perform parkour.
HVC295234: Philly parkour.
MED E008: Flash Mob Gathering
(a)
HVC523034: The police in New York shut down a small protest against a city council member's fundraiser
HVC288592: Education rally in the street.
HVC709059: People doing a choreographed dance for a campaign.
HVC562609: Footage and interviews from Egypt solidarity protests .
Egyptian Solidarity Demons... NYC, February 2011
HVC635692: A guy lets people cut off his dreads for charity .
HVC059444: Pro-union protest in Wisconsin , against the govenor .
HVC383369: A jump rope troup does double dutch.
(b)
HVC103253: Flash mob in Millennium Park, Chicago.
HVC508163: Faculty flash mob dance at high school pep rally.
HVC744324: People celebrate St. Patrick's Day with a parade in New York City .
HVC745339: Justin Bieber flash mob.
HVC824069: Amateur film of marathon.
HVC036225: Britney Spears flash mob.
HVC067623: Dance flash mob at Seattle public library .

FIGURE 6.5: Top ranked videos/images ordered left-to-right using (a) plain retrieval without reranking and (b) self-paced reranking. True/false labels are marked in the lower-right of every frame.

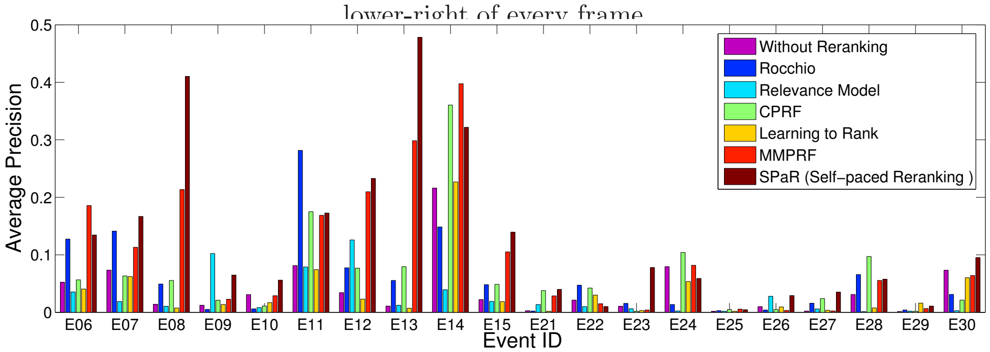
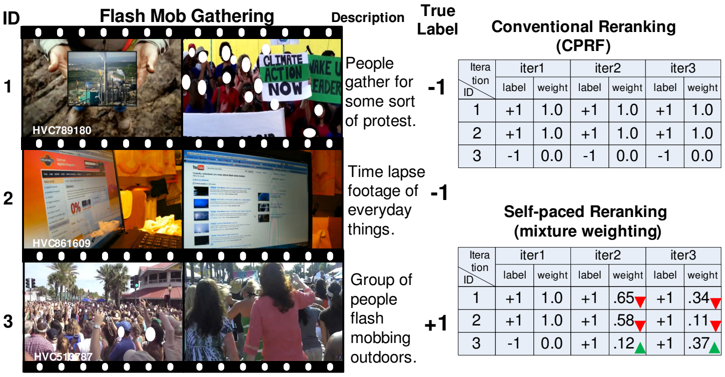


FIGURE 6.6: The AP comparison with the baseline methods. The MAP across all events is available in Table 6.1.

Fig. 6.7 illustrates the weights assigned by CPRF and SPaR on the event "E008 Flash Mob Gathering". Three representative videos are plotted where the third (ID=3) is true positive, and the others (ID=1,2) are negative. The tables on the right of Fig. 6.7 list their pseudo labels and weights in each iteration. Since the true labels are unknown to the methods, in the first iteration, both methods made mistakes. In Conventional Reranking, the initial pseudo labels and learned weights stay unchanged thereafter. However, SPaR adaptively adjusts their weights as the iteration grows, e.g. it reduces the overestimated weights of videos (ID=1,2) in iteration 2 and 3 probably because of their dissimilarity from other pseudo positive videos.

We found two scenarios where SPaR and MMPRF can fail. First, when the initial top-ranked samples retrieved by queries are completely off-topic. SPaR and MMPRF may not recover from the inferior starting values, e.g. the query brought by "E022 Cleaning an appliance" are off-topic (all videos are on all cooking in kitchen). Second, SPaR and MMPRF may not help when the features used in reranking are not discriminative to the queries, e.g. for "E025 Marriage Proposal", our system lacks of meaningful detectors


ID
Flash Mob Gathering
Description
True Label
1
HVC789180
People gather for some sort of protest.
-1
2
HVC861609
Time lapse footage of everyday things.
-1
3
HVC513787
Group of people flash mobbing outdoors.
+1

Conventional Reranking (CPRF)

Iteration / ID	iter1		iter2		iter3	
	label	weight	label	weight	label	weight
1	+1	1.0	+1	1.0	+1	1.0
2	+1	1.0	+1	1.0	+1	1.0
3	-1	0.0	-1	0.0	-1	0.0

Self-paced Reranking (mixture weighting)

Iteration / ID	iter1		iter2		iter3	
	label	weight	label	weight	label	weight
1	+1	1.0	+1	.65▼	+1	.34▼
2	+1	1.0	+1	.58▼	+1	.11▼
3	-1	0.0	+1	.12▲	+1	.37▲

FIGURE 6.7: Weights changed by CPRF and SPaR on representative videos in different iterations.

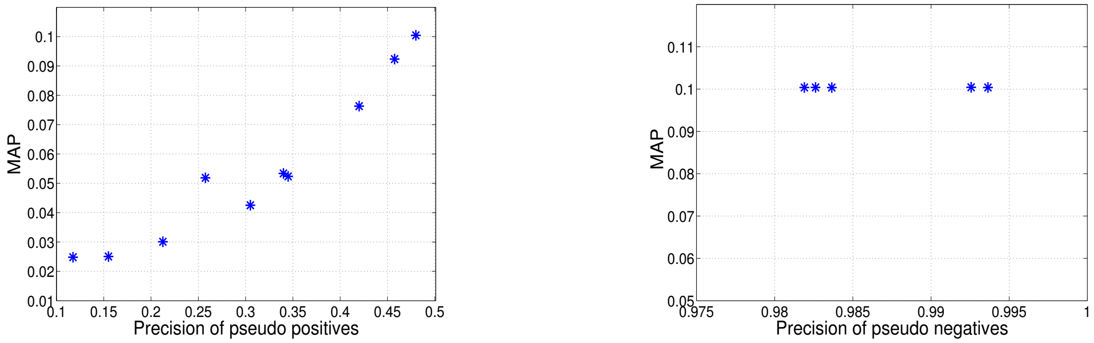
TABLE 6.1: MAP (× 100) comparison with the baseline methods across 20 Pre-Specified events.

Method	NIST's split
Without Reranking	3.9
Rocchio	5.7
Relevance Model	2.6
CPRF	6.4
Learning to Rank	3.4
MMPRF	10.1
SPaR	**12.9**

such as "stand on knees". Therefore even if 10 true positives are used, the AP is still bad.

6.5.3 Impact of Pseudo Label Accuracy

To study the impact of pseudo label accuracy, we conduct the following experiments, where the pseudo positive samples are simply selected by the top k^+ samples in the ranked lists of individual features and the fusion of all features. Figure 6.8(a) illustrates the result in a scatter plot where the x-axis represents the accuracy of pseudo positives and the y-axis represents the MAP. As can be seen, there is a strong correlation between the MAP and the accuracy of pseudo-positives. The average Pearson correlation is 0.93. We also conduct a similar experiment on pseudo negative samples, where the pseudo positive samples are fixed and the pseudo negative samples are randomly selected from the bottom of the initial ranked list. The experiments are conducted five times and the result is shown in Figure 6.8(b). As we see, the precision is always larger than 0.980 as false negatives are difficult to find. This observation agrees with the analytical result in Section 6.3. Given such highly accurate pseudo negatives, the impact of pseudo negatives on MAP seems to be marginal. In summary, the results demonstrate that the



(a) Pseudo positives

(b) Pseudo negatives

Figure 6.8: The correlation between pseudo label accuracy and MAP. Each point represents an experiment with pseudo samples with certain accuracy.

accuracy of pseudo positive samples has a substantial impact on the MAP. The impact of pseudo negative samples, however, appears to be negligible.

6.5.4 Comparison of Weighting Schemes

Section 6.4.2 discusses four weighting schemes including the conventional hard weighting and the proposed three soft weighting schemes. The following two predefined schemes are also included for comparison: 1) Interpolation is a commonly used weighting scheme which assigns weights linearly to a sample' rank order [115, 126]:

$$v_i = \frac{1}{m}\sum_{j=1}^{m}(1 - \frac{rank(\mathbf{x}_{ij})}{N}), \tag{6.24}$$

where N is the number of total pseudo samples. The weight for the first sample is 1.0, and 0.0 for the last. $rank(\cdot)$ returns the sample's rank order in its list. 2) Inverse Rank assigns a sample's weight based on its inverse rank order. The weight v_i equals the average inverse rank across m modalities:

$$v_i = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{rank(\mathbf{x}_{ij})}. \tag{6.25}$$

We conduct experiments with different weighting schemes and plot their MAPs in Fig. 6.9, where the x-axis denotes the iteration, and the y-axis is the MAP. The same step size is used in all methods. As we see, SPaR with the proposed soft weighting schemes, including linear, log and mixture weighting, outperforms the binary and the predefined weighting across iterations. Among them, the mixture weighting is slightly better than others, suggesting the rationale for tolerating small errors on this dataset. However, it needs an additional parameter to tune. The MAPs of the proposed soft weighting schemes seem to be robust and less sensitive to the iteration change. The

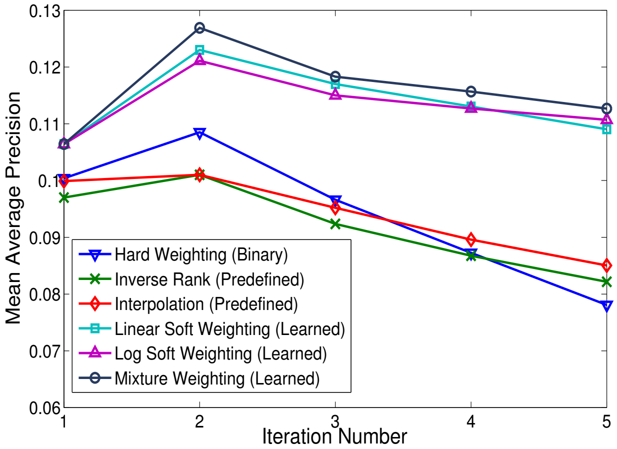


Figure 6.9: Comparison of binary, predefined, and learned weighting schemes in different iterations.

MAP drop, in all reranking methods, seems to be related to the nature of the MED dataset as the similar pattern can be observed in other reranking methods. Nevertheless, SPaR still outperforms the binary, predefined weights and the baseline methods in Table 6.1.

6.5.5 Experiments on Web Query Dataset

To verify SPaR's performance on image search, we conduct experiments on a web image query dataset consisting of 71,478 images from 353 queries, retrieved by a search engine named Exalead. For each query, the top ranked images generated by Exalead are provided, along with the true label for every image. The dataset is representative as the 353 queries cover a broad range of topics. The performance is evaluated by the non-interpolated MAP, as used in [111]. MAP@100 is also included for comparison. Note that as the initial result contains a single modality.

Following [112, 113], densely sampled SIFT are extracted. A codebook of 1,024 centroids is constructed. Spatial Tiling [43] is used to further improve the performance. We compare SPaR with the state-of-the-art reranking methods. SPaR is configured in a similar way as discussed in Section 6.5.1, and provided initial text-based search results are used. Following [112, 113], the parameters are tuned on a validation set consisting of a subset of 55 queries.

We examine the overall MAP in Table 6.2. "-" denotes that the number is unavailable in the cited paper. As we see, SPaR achieves the promising MAP among state-of-the-art reranking methods, including Graph-based [130], LETOR-based [112, 126], Classification-based [118] and even supervised reranking methods [111, 128], in terms of both MAP and MAP@100. A similar pattern can be observed that SPaR significantly

Table 6.2: MAP and MAP@100 comparison with baseline methods on the Web Query dataset.

Method	MAP	MAP@100
Without Reranking [111]	0.569	0.431
CPRF [118]	0.658	-
Random Walk [130]	0.616	-
Bayesian Reranking [112, 126]	0.658	0.529
Preference Learning Model [112]	-	0.534
BVLS [113]	0.670	-
Query-Relative(visual) [111]	0.649	-
Supervised Reranking [128]	0.665	-
SPaR	**0.672**	**0.557**

Table 6.3: Runtime Comparison in a single iteration.

Method	MED	Web Query
Rocchio	5.3 (s)	2.0 (s)
Relevance Model	7.2 (s)	2.5 (s)
Learning to Rank	178 (s)	22.3 (s)
CPRF	145 (s)	10.1 (s)
MMPRF	149 (s)	10.1 (s)
SPaR	158 (s)	12.2 (s)

boosts the MAP of plain retrieval without reranking, and obtain comparable or even better performance than the baseline methods. Generally, SPaR improves about 84% queries over the method without reranking. Since the initial ranked lists are retrieved by text matching, this result substantiates the claim that SPaR is general and applicable to reranking by text-based search.

6.5.6 Runtime Comparison

To empirically verify the efficiency of SPaR and MMPRF, we compare the runtime (second/query) in a single iteration. The experiments are conducted on Intel Xeon E5649 @ 2.53GHz with 16GB memory and the results are listed in Table 6.3. To test the speed of Rocchio and Relevance Model, we built our own inverted index on the Web Query dataset, and issue the query against the index. The reranking in MED, which is conducted on semantic features, is slower because it involves multiple features and modalities. As we see, SPaR's overhead over CPRF is marginal on the both sets. This result suggests SPaR and MMPRF is inexpensive. Note the implementations for all methods reported here are far from optimal, which involve a number of programming languages. We will report the runtime of the accelerated pipeline in Section 8.

6.6 Summary

In this chapter, we proposed two approaches for multimodal reranking, namely Multi-Modal Pseudo Relevance Feedback (MMPRF) and Self-Paced Reranking (SPaR). Unlike existing methods, the reranking is conducted using multiple ranked lists. In MMPRF, we formulated the pseudo label construction problem as maximum likelihood estimation and maximum expected value estimation problems, which can be solved by existing linear/integer programming algorithms. By training a joint model on the pseudo label set, MMPRF leverages low-level features and high-level features for multimedia event detection without any training data. SPaR reveals the link between reranking and an optimization problem that can be effectively solved by self-paced learning. The proposed SPaR is general, and can be used to theoretically explain other reranking methods including MMPRF. Experimental results validate the efficacy and the efficiency of the proposed methods on several datasets. The proposed methods consistently outperforms the plain retrieval without reranking, and obtains decent improvements over existing reranking methods.

The quality of the initial feedback samples in the top ranked results affects the reranking performance. Generally, reranking improves the accuracy of some queries whereas hurts the accuracy of some queries, and in the end, improves the average accuracy across all queries. Admittedly, in some cases where the initial feedback samples are of low quality, reranking may not help. In this case, we recommend employing supervised reranking algorithms [111, 128], in which human supervision can be used to either 1) obtain feedback samples with higher quality, or 2) select the queries that might benefit from reranking [125]. Supervised reranking is out of the scope of this thesis, and we refer readers to relevant papers.

Chapter 7

Learning Semantic Concepts

7.1 Introduction

Concept detectors is the key in a CBVSR system as it affects what can be searched by semantic search. Concept detectors can be trained on still images or videos. The latter approach is more desirable due to 1) a minimal domain difference and 2) capability for action, and 3) possibility for audio detection. In this chapter, we explore a semantic concept learning method using self-paced curriculum learning. The theory has been used in the reranking method SPaR in Chapter 6.4. In this chapter, we will formally introduce the general form of the theory and discuss its application on semantic concept training. We approach this problem based on recently proposed theories called *curriculum learning* [119] and *self-paced learning* [120]. The theories have been attracting increasing attention in the field of machine learning and artificial intelligence. Both the learning paradigms are inspired by the learning principle underlying the cognitive process of humans and animals, which generally start with learning easier aspects of a task, and then gradually take more complex examples into consideration. The intuition can be explained in analogous to human education in which a pupil is supposed to understand elementary algebra before he or she can learn more advanced algebra topics. This learning paradigm has been empirically demonstrated to be instrumental in avoiding bad local minima and in achieving a better generalization result [140–142].

A curriculum determines a sequence of training samples which essentially corresponds to a list of samples ranked in ascending order of learning difficulty. A major disparity between *curriculum learning* (CL) and *self-paced learning* (SPL) lies in the derivation of the curriculum. In CL, the curriculum is assumed to be given by an oracle beforehand, and remains fixed thereafter. In SPL, the curriculum is dynamically generated by the learner itself, according to what the learner has already learned.

The advantage of CL includes the flexibility to incorporate prior knowledge from various sources. Its drawback stems from the fact that the curriculum design is determined independently of the subsequent learning, which may result in inconsistency between the fixed curriculum and the dynamically learned models. From the optimization perspective, since the learning proceeds iteratively, there is no guarantee that the predetermined curriculum can even lead to a converged solution. SPL, on the other hand, formulates the learning problem as a concise biconvex problem, where the curriculum design is embedded and jointly learned with model parameters. Therefore, the learned model is consistent. However, SPL is limited in incorporating prior knowledge into learning, rendering it prone to overfitting. Ignoring prior knowledge is less reasonable when reliable prior information is available. Since both methods have their advantages, it is difficult to judge which one is better in practice.

In this chapter, we discover the missing link between CL and SPL. We formally propose a unified framework called *Self-paced Curriculum Leaning* (SPCL). SPCL represents a general learning paradigm that combines the merits from both the CL and SPL. On one hand, it inherits and further generalizes the theory of SPL. On the other hand, SPCL addresses the drawback of SPL by introducing a flexible way to incorporate prior knowledge. This chapter offers a compelling insight on the relationship between the existing CL and SPL methods. Their relation can be intuitively explained in the context of human education, in which SPCL represents an "instructor-student collaborative" learning paradigm, as opposed to "instructor-driven" in CL or "student-driven" in SPL. In SPCL, instructors provide prior knowledge on a weak learning sequence of samples, while leaving students the freedom to decide the actual curriculum according to their learning pace. Since an optimal curriculum for the instructor may not necessarily be optimal for all students, we hypothesize that given reasonable prior knowledge, the curriculum devised by instructors and students together can be expected to be better than the curriculum designed by either part alone.

7.2 Related Work

7.2.1 Curriculum Learning

Bengio et al. proposed a new learning paradigm called *curriculum learning* (CL), in which a model is learned by gradually including from easy to complex samples in training so as to increase the entropy of training samples [119]. Afterwards, Bengio and his colleagues presented insightful explorations for the rationality underlying this learning paradigm, and discussed the relationship between CL and conventional optimization techniques, e.g., the continuation and annealing methods [143, 144]. From human behavioral perspective, evidence have shown that CL is consistent with the principle in human teaching [140, 141].

The CL methodology has been applied to various applications, the key in which is to find a ranking function that assigns learning priorities to training samples. Given a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where $\mathbf{x}_i$ denotes the i^{th} observed sample, and y_i represents its label. A curriculum is characterized by a ranking function γ. A sample with a higher rank, i.e., smaller value, is supposed to be learned earlier.

The curriculum (or the ranking function) is often derived by predetermined heuristics for particular problems. For example, in the task of classifying geometrical shapes, the ranking function was derived by the variability in shape [119]. The shapes exhibiting less variability are supposed to be learned earlier. In [140], the authors tried to teach a robot the concept of "graspability" - whether an object can be grasped and picked up with one hand, in which participants were asked to assign a learning sequence of graspability to various object. The ranking is determined by common sense of the participants. In [145], the authors approached grammar induction, where the ranking function is derived in terms of the length of a sentence. The heuristic is that the number of possible solutions grows exponentially with the length of the sentence, and short sentences are easier and thus should be learn earlier.

The heuristics in these problems turn out to be beneficial. However, the heuristical curriculum design may lead to inconsistency between the fixed curriculum and the dynamically learned models. That is, the curriculum is predetermined a priori and cannot be adjusted accordingly, taking into account the feedback about the learner.

7.2.2 Self-paced Learning

To alleviate the issue of CL, Koller's group [120] designed a new formulation, called *self-paced learning* (SPL). SPL embeds curriculum design as a regularization term into the learning objective. Compared with CL, SPL exhibits two advantages: first, it jointly optimizes the learning objective together with the curriculum, and therefore the curriculum and the learned model are consistent under the same optimization problem; second, the regularization term is independent of loss functions of specific problems. This theory has been successfully applied to various applications, such as action/event detection [77], reranking [15], domain adaption [142], dictionary learning [146], tracking [147] and segmentation [148].

Formally, let $L(y_i, g(\mathbf{x}_i, \mathbf{w}))$ denote the loss function which calculates the cost between the ground truth label y_i and the estimated label $g(\mathbf{x}_i, \mathbf{w})$. Here $\mathbf{w}$ represents the model parameter inside the decision function g. In SPL, the goal is to jointly learn the model parameter $\mathbf{w}$ and the latent weight variable $\mathbf{v} = [v_1, \cdots, v_n]^T$ by minimizing:

$$\min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v};\lambda) = \sum_{i=1}^{n} v_i L(y_i, g(\mathbf{x}_i, \mathbf{w})) - \lambda \sum_{i=1}^{n} v_i, \tag{7.1}$$

where λ is a parameter for controlling the learning pace. Eq. (7.1) indicates the loss of a sample is discounted by a weight. The objective of SPL is to minimize the weighted training loss together with the negative l_1-norm regularizer $-\|\mathbf{v}\|_1 = -\sum_{i=1}^{n} v_i$ (since $v_i \geq 0$). A more general regularizer consists of both $\|\mathbf{v}\|_1$ and the sum of group-wise $\|\mathbf{v}\|_2$ [77].

ACS (Alternative Convex Search) is generally used to solve Eq. (7.1) [133]. ACS is a special case of Cyclic Coordinate Method (CCM) [133] discussed in Section 6.4. It is an iterative method for biconvex optimization, in which the variables are divided into two disjoint blocks. In each iteration, a block of variables are optimized while keeping the other block fixed. With the fixed $\mathbf{w}$, the global optimum $\mathbf{v}^* = [v_1^*, \cdots, v_n^*]$ can be easily calculated by:

$$v_i^* = \begin{cases} 1, & L(y_i, g(\mathbf{x}_i, \mathbf{w})) < \lambda, \\ 0, & \text{otherwise.} \end{cases} \tag{7.2}$$

There exists an intuitive explanation behind this alternative search strategy: first, when updating $\mathbf{v}$ with a fixed $\mathbf{w}$, a sample whose loss is smaller than a certain threshold λ is taken as an "easy" sample, and will be selected in training ($v_i^* = 1$), or otherwise unselected ($v_i^* = 0$); second, when updating $\mathbf{w}$ with a fixed $\mathbf{v}$, the classifier is trained only on the selected "easy" samples. The parameter λ controls the pace at which the model learns new samples, and physically λ corresponds to the "age" of the model. When λ is small, only "easy" samples with small losses will be considered. As λ grows, more samples with larger losses will be gradually appended to train a more "mature" model.

This strategy complies with the heuristics in most CL methods [119, 140]. However, since the learning is completely dominated by the training loss, the learning may be prone to overfitting. Moreover, it provides no way to incorporate prior guidance in learning. To the best of our knowledge, there has been no studies to incorporate prior knowledge into SPL, nor to analyze the relation between CL and SPL.

7.2.3 Weakly-Labeled Data Learning

Recently, a few studies have been proposed trying to utilize the huge amount of noisy data from the Internet. For example, Mitchell et al. [149] proposed a Never-Ending

Language Learning (NELL) paradigm and built adaptive learners that makes use of the web data by learning different types of knowledge and beliefs continuously. Such learning process is mostly self-supervised, and previously learned knowledge enables learning further types of knowledge. Sukhbaatar et al. [150] designed loss layers specifically for noisy label learning of images in Convolutional Neural Network. It tried to estimate the distribution of noise and was mainly verified on synthesized noisy labels. Liang et al. [151] presented a weakly-supervised method called Baby Learning for object detection from a few training images and videos. They first embed the prior knowledge into a pre-trained CNN. When given very few samples for a new concept, a simple detector is constructed to discover much more training instances from the online weakly labeled videos. As more training samples are selected, the concept detector keeps refining until a mature detector is formed. Varadarajan et al. [48] discussed a method that exploits the YouTube topic API to train large scale video concept detectors on YouTube. The method utilized a calibration process and hard negative mining to train a second order mixture of experts model in order to discover correlations within the labels. Existing methods are mainly built on heuristic approaches and it is unclear what objective is being optimized. In this paper, we theoretically justify the proposed method and empirically demonstrate its superior performance over representative existing methods.

7.3 Theory

An ideal learning paradigm should consider both prior knowledge known before training and information learned during training in a unified and sound framework. Similar to human education, we are interested in constructing an "instructor-student collaborative" paradigm, which, on one hand, utilizes prior knowledge provided by instructors as a guidance for curriculum design (the underlying CL methodology), and, on the other hand, leaves students certain freedom to adjust to the actual curriculum according to their learning paces (the underlying SPL methodology). This requirement can be realized through the following optimization model. Similar in CL, we assume that the model is given a curriculum that is predetermined by an oracle. Following the notation defined above, we have:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w},\mathbf{v};\lambda,\Psi) = \sum_{i=1}^{n} v_i L(y_i, g(\mathbf{x}_i,\mathbf{w})) + f(\mathbf{v};\lambda) \\ \text{s.t. } \mathbf{v} \in \Psi \end{aligned} \tag{7.3}$$

where $\mathbf{v} = [v_1, v_2, \cdots, v_n]^T$ denote the weight variables reflecting the samples' importance. f is called self-paced function which controls the learning scheme; Ψ is a feasible region that encodes the information of a predetermined curriculum.

SPCL represents a general learning framework which includes CL and SPL as special cases. SPCL degenerates to SPL when the curriculum region is ignored ($\Psi = [0,1]^n$), or equivalently, the prior knowledge on predefined curriculums is absent. In this case, the learning is totally driven by the learner. SPCL degenerates to CL when the curriculum region (feasible region) only contains the learning sequence in the predetermined curriculum. In this case, the learning process neglects the feedback about learners, and is dominated by the given prior knowledge. When information from both sources are available, the learning in SPCL is collaboratively driven by prior knowledge and learning objective. Table 7.1 summarizes the characteristics of different learning methods. Given reasonable prior knowledge, SPCL which considers the information from both sources tend to yield better solutions. Example 7.1 shows a case in this regard.

7.3.1 Curriculum

A curriculum can be mathematically described as:

Definition 7.1 (Total order curriculum)**.** For training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, a total order curriculum, or curriculum for short, can be expressed as a ranking function:

$$\gamma : \mathbf{X} \rightarrow \{1, 2, \cdots, n\},$$

where $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$ represents that x_i should be learned earlier than x_j in training. $\gamma(\mathbf{x}_i) = \gamma(\mathbf{x}_j)$ denotes there is no preferred learning order on the two samples.

Definition 7.2 (Curriculum region)**.** Given a predetermined curriculum $\gamma(\cdot)$ on training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ and their weight variables $\mathbf{v} = [v_1, \cdots, v_n]^T$. A feasible region Ψ is called a curriculum region of γ if

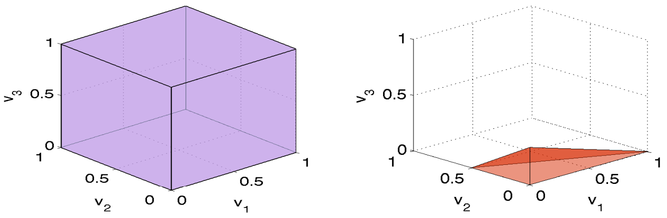
1. Ψ is a nonempty convex set;
2. for any pair of samples $\mathbf{x}_i, \mathbf{x}_j$, if $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$, it holds that $\int_\Psi v_i \, d\mathbf{v} > \int_\Psi v_j \, d\mathbf{v}$, where $\int_\Psi v_i \, d\mathbf{v}$ calculates the expectation of v_i within Ψ. Similarly if $\gamma(\mathbf{x}_i) = \gamma(\mathbf{x}_j)$, $\int_\Psi v_i \, d\mathbf{v} = \int_\Psi v_j \, d\mathbf{v}$.

The two conditions in Definition 7.2 offer a realization for curriculum learning. Condition 1 ensures the soundness for calculating the constraints. Condition 2 indicates that samples to be learned earlier should have larger expected values. The curriculum region physically corresponds to a convex region in the high-dimensional space. The area inside this region confines the space for learning the weight variables. The shape of the region weakly implies a prior learning sequence of samples, where the expected values for favored samples are larger. For example, Figure 7.1(b) illustrates an example of feasible region in 3D where the x, y, z axis represents the weight variable v_1, v_2, v_3,

Table 7.1: Comparison of different learning approaches.

	CL	**SPL**	**Proposed SPCL**
Comparable to human learning	Instructor-driven	Student-driven	Instructor-student collaborative
Curriculum design	Prior knowledge	Learning objective	Learning objective + prior knowledge
Learning schemes	Multiple	Single	Multiple
Iterative training	Heuristic approach	Gradient-based	Gradient-based

respectively. Without considering the learning objective, we can see that v_1 tends to be learned earlier than v_2 and v_3. This is because if we uniformly sample sufficient points in the feasible region of the coordinate (v_1, v_2, v_3), the expected value of v_1 is larger. Since prior knowledge is missing in Eq. (7.1), the feasible region is a unit hypercube, i.e. all samples are equally favored, as shown in Figure 7.1(a). Note the curriculum region should be confined within the unit hypercube since the constraints $\mathbf{v} \in [0, 1]^n$ in Eq. (7.3).



(a) SPL (b) SPCL

Figure 7.1: Comparison of feasible regions in SPL and SPCL.

Note that the prior learning sequence in the curriculum region only weakly affects the actual learning sequence, and it is very likely that the prior sequence will be adjusted by the learners. This is because the prior knowledge determines a weak ordering of samples that suggests what should be learned first. A learner takes this knowledge into account, but has his/her own freedom to alter the sequence in order to adjust to the learning objective. See Example 7.1. Therefore, SPCL represents an "instructor-student-corporative" learning paradigm.

Predetermining a total-order learning sequence for every pair of samples, especially in big data, seems to be infeasible in many problems. In reality, we can only obtain incomplete prior information from the noisy data. For examples, we may know some videos with certain keywords in its title should be learned earlier, but may never know the learning priority for the videos that do not have the keywords. To this end, we also propose partial-order curriculum, which allows for leveraging the incomplete prior information. Define a partial order relation $\preceq$ such that $x_i \preceq x_j$ indicates that the sample x_i should be learned no later than x_j $(i, j \in [1, n])$. Similarly given two sample subsets $\mathbf{X}_a \preceq \mathbf{X}_b$ denotes the samples in $\mathbf{X}_a$ should be learned no later than the samples in $\mathbf{X}_b$.

Definition 7.3 (Partial order curriculum)**.** Given the training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ and their weight variables $\mathbf{v} = [v_1, \cdots, v_n]^T$. Define a partial-order set $\gamma = (\mathbf{X}, \preceq)$. For every element in set $\mathbf{X}_p \preceq \mathbf{X}_q(\mathbf{X}_p, \mathbf{X}_q \subseteq \mathbf{X})$, a feasible region $\Psi = (\mathbf{A}^T\mathbf{v} \leq 0)$ is called a partial-order curriculum region of γ if

1. $\mathbf{A} = 0$ is a zero matrix except
2. $\forall x_i \in \mathbf{X}_p$, $\forall x_j \in \mathbf{X}_q$ we have $\exists t$, $\mathbf{A}_{ti} = -1$ and $\mathbf{A}_{tj} = 1$. Otherwise .

Definition 7.3 incorporates the constraints on groups of examples as opposed to every pair of examples. In other words, for the partial order $\mathbf{x}_i \preceq \mathbf{x}_j$, it holds that $\int_\Psi v_i \, d\mathbf{v} > \int_\Psi v_j \, d\mathbf{v}$, where $\int_\Psi v_i \, d\mathbf{v}$ calculates the expectation of v_i within Ψ. And if $\mathbf{x}_i \preceq \mathbf{x}_j$ and $\mathbf{x}_j \preceq \mathbf{x}_i$, it holds $\int_\Psi v_i \, d\mathbf{v} = \int_\Psi v_j \, d\mathbf{v}$.

The partial-order curriculum in Definition 7.3 generalizes the total-order curriculum Definition 7.1 by incorporating the incomplete prior over groups of samples. Samples in the confident groups should be learned earlier than samples in the less confident groups. It imposes no prior over the samples within the same group nor the samples not in any group. Definition **??** follows the curriculum definition in [78] and will degenerate to the curriculum in [78] when the partial order becomes the full order relation.

7.3.2 Self-pace Functions

Compared with Eq. (7.1), SPCL generalizes SPL by introducing a regularization term. This term determines the learning scheme, i.e., the strategy used by the model to learn new samples. In human learning, we tend to use different schemes for different tasks. Similarly, SPCL should also be able to utilize different learning schemes for different problems. Since the existing methods only include a single learning scheme, we generalize the learning scheme and define:

Definition 7.4 (Self-paced function)**.** A self-paced function determines a learning scheme. Suppose that $\mathbf{v} = [v_1, \cdots, v_n]^T$ denotes a vector of weight variable for each training sample and $\ell = [\ell_1, \cdots, \ell_n]^T$ are the corresponding loss. λ controls the learning pace (or model "age"). $f(\mathbf{v}; \lambda)$ is called a self-paced function, if

1. $f(\mathbf{v}; \lambda)$ is convex with respect to $\mathbf{v} \in [0, 1]^n$.
2. When all variables are fixed except for v_i and ℓ_i, v_i^* decreases with ℓ_i, and it holds that $\lim\limits_{\ell_i \to 0} v_i^* = 1, \lim\limits_{\ell_i \to \infty} v_i^* = 0$.
3. $\|\mathbf{v}\|_1 = \sum_{i=1}^n v_i$ increases with respect to λ, and it holds that $\forall i \in [1, n]$, $\lim\limits_{\lambda \to 0} v_i^* = 0, \lim\limits_{\lambda \to \infty} v_i^* = 1$.

where $\mathbf{v}^* = \arg\min_{\mathbf{v} \in [0,1]^n} \sum v_i \ell_i + f(\mathbf{v}; \lambda)$, and denote $\mathbf{v}^* = [v_1^*, \cdots, v_n^*]$.

The three conditions in Definition 7.4 provide a definition for the self-paced learning scheme. Condition 2 indicates that the model inclines to select easy samples (with smaller losses) in favor of complex samples (with larger losses). Condition 3 states that when the model "age" λ gets larger, it should incorporate more, probably complex, samples to train a "mature" model. The convexity in Condition 1 ensures the model can find good solutions within the curriculum region.

It is easy to verify that the regularization term in Eq. (7.1) satisfies Definition 7.4. In fact, this term corresponds to a binary learning scheme since v_i can only take binary values, as shown in the closed-form solution of Eq. (7.2). This scheme may be less appropriate in the problems where the importance of samples needs to be discriminated. In fact, there exist a plethora of self-paced functions corresponding to various learning schemes. We will detail some of them in the next section.

7.3.3 Algorithm

Inspired by the algorithm in [120], we employ a similar ACS algorithm to solve Eq. (7.3). Algorithm 2 takes the input of a predetermined curriculum, an instantiated self-paced function and a stepsize parameter; it outputs an optimal model parameter $\mathbf{w}$. First of all, it represents the input curriculum as a curriculum region that follows Definition 2, and initializes variables in their feasible region. Then it alternates between two steps until it finally converges: Step 4 learns the optimal model parameter with the fixed and most recent $\mathbf{v}^*$; Step 5 learns the optimal weight variables with the fixed $\mathbf{w}^*$. In first several iterations, the model "age" is increased so that more complex samples will be gradually incorporated in the training. For example, we can increase λ so that μ more samples will be added in the next iteration. According to the conditions in Definition 7.4, the number of complex samples increases along with the growth of the number iteration. Step 4 can be conveniently implemented by existing off-the-shelf supervised learning methods. Gradient-based or interior-point methods can be used to solve the convex optimization problem in Step 5. According to [133], the alternative search in Algorithm 2 converges as the objective function is monotonically decreasing and is bounded from below.

7.4 Implementation

The definitions discussed above provide a theoretical foundation for SPCL. However, we still need concrete self-paced functions and curriculum regions to solve specific problems. To this end, this section discusses some implementations that follow Definition 7.2 and Definition 7.4. Note that there is no single implementation that can always work the

Algorithm 2: Self-paced Curriculum Learning.

```
input  : Input dataset $\mathcal{D}$, predetermined curriculum $\gamma$, self-paced function $f$ and a
         stepsize $\mu$
output: Model parameter $\mathbf{w}$
Derive the curriculum region $\Psi$ from $\gamma$;
Initialize $\mathbf{v}^*$, $\lambda$ in the curriculum region;
while not converged do
    Update $\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathbb{E}(\mathbf{w}, \mathbf{v}^*; \lambda, \Psi)$;
    Update $\mathbf{v}^* = \arg\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}^*, \mathbf{v}; \lambda, \Psi)$;
    if $\lambda$ is small then increase $\lambda$ by the stepsize $\mu$;
end
return $\mathbf{w}^*$
```

best for all problems. Our purpose is to argument the implementations in the literature, and to help enlighten others to further explore this interesting direction.

7.4.1 Curriculum region implementation

We suggest an implementation induced from a linear constraint for realizing the curriculum region: $\mathbf{a}^T\mathbf{v} \leq c$, where $\mathbf{v} = [v_1, \cdots, v_n]^T$ are the weight variables in Eq. (7.3), c is a constant, and $\mathbf{a} = [a_1, \cdots, a_n]^T$ is a n-dimensional vector. The linear constraints is a simple implementation for curriculum region that can be conveniently solved. It can be proved that this implementation complies with the definition of curriculum region.

Theorem 7.5. *For training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, given a curriculum γ defined on it, the feasible region, defined by,*

$$\Psi = \{\mathbf{v} | \mathbf{a}^T\mathbf{v} \leq c\}$$

is a curriculum region of γ if it holds: 1) $\Psi \wedge \mathbf{v} \in [0,1]^n$ is nonempty; 2) $a_i < a_j$ for all $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$; $a_i = a_j$ for all $\gamma(\mathbf{x}_i) = \gamma(\mathbf{x}_j)$.

We also suggest an implementation for partial order curriculum in the context of video concept learning. We can derive the partial-order curriculum in the following way: we only distinguish the training order for groups of samples. We directly utilize the textual descriptions of the videos generated by the uploaders. For each video, we extract the latent topics of the video based on their titles, descriptions and tags in their metadata. In terms of the distance between the video's latent topic to the target concept, we group videos in a sequential order for each concept. The grouping and ordering information of the videos can be used to construct the partial-order curriculum. In our experiment, we divide the data into two partial-order curriculum groups, where the videos with matching scores larger than zero are in one group and the rest are in the other group.

7.4.2 Self-paced function implementation

Similar to the scheme human used to absorb knowledge, a self-paced function determines a learning scheme for the model to learn new samples. Note the self-paced function is realized as a regularization term, which is independent of specific loss functions, and can be easily applied to various problems. Since human tends to use different learning schemes for different tasks, SPCL should also be able to utilize different learning schemes for different problems. Inspired by a study in [15], this section discusses some examples of learning schemes.

Binary scheme: This scheme in is used in [120]. It is called binary scheme, or "hard" scheme, as it only yields binary weight variables.

$$f(\mathbf{v};\lambda) = -\lambda\|v\|_1 = -\lambda\sum_{i=1}^{n} v_i, \tag{7.4}$$

Linear scheme: A common approach is to linearly discriminate samples with respect to their losses. This can be realized by the following self-paced function:

$$f(\mathbf{v};\lambda) = \frac{1}{2}\lambda\sum_{i=1}^{n}(v_i^2 - 2v_i), \tag{7.5}$$

in which $\lambda > 0$. This scheme represents a "soft" scheme as the weight variable can take real values.

Logarithmic scheme: A more conservative approach is to penalize the loss logarithmically, which can be achieved by the following function:

$$f(\mathbf{v};\lambda) = \sum_{i=1}^{n}\zeta v_i - \frac{\zeta^{v_i}}{\log\zeta}, \tag{7.6}$$

where $\zeta = 1 - \lambda$ and $0 < \lambda < 1$.

Mixture scheme: Mixture scheme is a hybrid of the "soft" and the "hard" scheme [15]. If the loss is either too small or too large, the "hard" scheme is applied. Otherwise, the soft scheme is applied. Compared with the "soft" scheme, the mixture scheme tolerates small errors up to a certain point. To define this starting point, an additional parameter is introduced, i.e. $\lambda = [\lambda_1, \lambda_2]^T$. Formally,

$$f(\mathbf{v};\lambda) = -\zeta\sum_{i=1}^{n}\log(v_i + \frac{1}{\lambda_1}\zeta), \tag{7.7}$$

where $\zeta = \frac{\lambda_1\lambda_2}{\lambda_1-\lambda_2}$ and $\lambda_1 > \lambda_2 > 0$.

Theorem 7.6. *The binary, linear, logarithmic and mixture scheme function are self-paced functions.*

It can be proved that the above functions follow Definition 7.4. The name of the learning scheme suggests the characteristic of its solution. The curve in Fig. 6.4 illustrates the characteristics of the learning schemes. When the curriculum region is not a unit hypercube, the closed-form solution, such as Eq. (7.2) cannot be directly used. Gradient-based methods can be applied. As $\mathbb{E}_{\mathbf{w}}$ is convex, the local optimal is also the global optimal solution for the subproblem.

Example 7.1. *Given six samples a, b, c, d, e, f. In the current iteration, the losses for these samples are $\ell = [0.1, 0.2, 0.4, 0.6, 0.5, 0.3]$, respectively. A latent ground-truth curriculum is listed in the first row of the following table, followed by the curriculum of CL, SPL and SPCL. For simplicity, binary scheme is used in SPL and SPCL where $\lambda = 0.8333$. If two samples with the same weight, we rank them in ascending order of their losses, in order to break the tie. The Kendall's rank correlation is presented in the last column.*

Method	*Curriculum*	*Correlation*
Ground-Truth	a, b, c, d, e, f	-
CL	b, a, d, c, e, f	*0.73*
SPL	a, b, f, c, e, d	*0.46*
SPCL	a, b, c, d, e, f	*1.00*

The curriculum region used is a linear constraint $\mathbf{a}^T\mathbf{v} \leq 1$, where $\mathbf{a} = [0.1, 0.0, 0.4, 0.3, 0.5, 1.0]^T$. In the implementation, we add a small constant 10^{-7} in the constraints for optimization accuracy. The constraint follows Definition 2 in the paper. As shown, both CL and SPL yield the suboptimal curriculum, e.g. their correlations are only 0.73 and 0.46. However, SPCL exploits the complementary information in CL and SPL, and devises an optimal curriculum. Note that CL recommends to learn b before a, but SPCL disobeys this order in the actual curriculum. The final solution of SPCL is $\mathbf{v}^* = [1.00, 1.00, 1.00, 0.88, 0.47, 0.00]$.

When the predetermined curriculum is completely wrong, SPCL may still be robust to the inferior prior knowledge given reasonable curriculum regions are applied. In this case, the prior knowledge should not be encoded as strong constraints. For example, in the above example, we can use the following curriculum region to encode the completely incorrect predetermined curriculum: $\mathbf{a}^T\mathbf{v} \leq 6.0$, where $\mathbf{a} = [2.3, 2.2, 2.1, 2.0, 1.7, 1.5]^T$

Method	*Curriculum*	*Correlation*
CL	f, e, d, c, b, a	*-1.00*
SPL	a, b, f, c, e, d	*0.46*
SPCL	a, f, b, c, e, d	*0.33*

As we see, even though the predetermined curriculum is completely wrong (correlation -1.00), the proposed SPCL still obtains reasonable curriculum (correlation 0.33). This *is because SPCL is able to leverage information in both prior knowledge and learning objective. The optimal solution of SPCL is* $\mathbf{v}^* = [1.00, 0.91, 0.10, 0.00, 0.00, 1.00]$.

7.4.3 Self-paced function with diversity

In the above learning schemes, samples in a curriculum are selected solely in terms of "easiness". In this section, we reveal that diversity, an important aspect in learning, should also be considered. Ideal learning should utilize not only easy but also diverse examples that are sufficiently dissimilar from what has already been learned. This can be intuitively explained in the context of human education. A rational curriculum for a pupil not only needs to include examples of suitable easiness matching her learning pace, but also, importantly, should include some diverse examples on the subject in order for her to develop more comprehensive knowledge. Likewise, learning from easy and diverse samples is expected to be better than learning from either criterion alone. To this end, we propose the following learning scheme.

Diverse learning scheme: Diversity implies that the selected samples should be less similar or clustered. An intuitive approach for realizing this is by selecting samples of different groups scattered in the sample space. We assume that the correlation of samples between groups is less than that of within a group. This auxiliary group membership is either given, e.g. in object recognition frames from the same video can be regarded from the same group, or can be obtained by clustering samples.

This aim can be mathematically described as follows. Assume that the training samples $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n) \in \mathbb{R}^{m \times n}$ are partitioned into b groups: $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(b)}$, where columns of $\mathbf{X}^{(j)} \in \mathbb{R}^{m \times n_j}$ correspond to the samples in the j^{th} group, n_j is the sample number in the group and $\sum_{j=1}^{b} n_j = n$. Accordingly denote the weight vector as $\mathbf{v} = [\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(b)}]$, where $\mathbf{v}^{(j)} = (v_1^{(j)}, \cdots, v_{n_j}^{(j)})^T \in [0,1]^{n_j}$. The diverse learning scheme on one hand needs to assign nonzero weights of $\mathbf{v}$ to easy samples as the hard learning scheme, and on the other hand requires to disperse nonzero elements across possibly more groups $\mathbf{v}^{(i)}$ to increase the diversity. Both requirements can be uniformly realized through the following optimization model:

$$\min_{\mathbf{w},\mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda, \gamma) = \sum_{i=1}^{n} v_i L(y_i, f(\mathbf{x}_i, \mathbf{w})) - \lambda \sum_{i=1}^{n} v_i - \gamma \|\mathbf{v}\|_{2,1}, \text{ s.t. } \mathbf{v} \in [0,1]^n, \tag{7.8}$$

where λ, γ are the parameters imposed on the easiness term (the negative l_1-norm: $-\|\mathbf{v}\|_1$) and the diversity term (the negative $l_{2,1}$-norm: $-\|\mathbf{v}\|_{2,1}$), respectively. As for the diversity term, we have:

$$-\|\mathbf{v}\|_{2,1} = -\sum_{j=1}^{b} \|\mathbf{v}^{(j)}\|_2. \tag{7.9}$$

The new regularization term consists of two components. One is the negative l_1-norm inherited from the hard learning scheme in SPL, which favors selecting easy over complex examples. The other is the proposed negative $l_{2,1}$-norm, which favors selecting diverse samples residing in more groups. It is well known that the $l_{2,1}$-norm leads to the group-wise sparse representation of $\mathbf{v}$ [71], i.e. non-zero entries of $\mathbf{v}$ tend to be concentrated in a small number of groups. Contrariwise, the negative $l_{2,1}$-norm should have a counter-effect to group-wise sparsity, i.e. nonzero entries of $\mathbf{v}$ tend to be scattered across a large number of groups. In other words, this anti-group-sparsity representation is expected to realize the desired diversity. Note that when each group only contains a single sample, Eq. (A.16) degenerates to Eq. (7.1).

Unlike the convex regularization term above, the term in diverse learning scheme is non-convex. A challenge is optimizing $\mathbf{v}$ with a fixed $\mathbf{w}$ becoming a non-convex problem. To this end, we propose a simple yet effective algorithm for extracting the global optimum of this problem when the curriculum is as in SPL, i.e. $\Psi = [0,1]^n$. Algorithm 3 takes as input the groups of samples, the up-to-date model parameter $\mathbf{w}$, and two self-paced parameters, and outputs the optimal $\mathbf{v}$ of $\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda, \gamma)$. The global minimum is proved in Appendix A:

Theorem 7.7. *Algorithm 3 attains the global optimum to* $\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v})$ *for any given* $\mathbf{w}$ *in linearithmic time.*

Algorithm 3: Algorithm for Solving $\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda, \gamma)$.

```
  input  : Input dataset $\mathcal{D}$, groups $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(b)}$, $\mathbf{w}$, $\lambda$, $\gamma$
  output: The global solution $\mathbf{v} = (\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(b)})$ of $\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda, \gamma)$.
1 for $j = 1$ to $b$ do // for each group
2     Sort the samples in $\mathbf{X}^{(j)}$ as $(\mathbf{x}_1^{(j)}, \cdots, \mathbf{x}_{n_j}^{(j)})$ in ascending order of their loss values $L$;
3     Accordingly, denote the labels and weights of $\mathbf{X}^{(j)}$ as $(y_1^{(j)}, \cdots, y_{n_j}^{(j)})$ and $(v_1^{(j)}, \cdots, v_{n_j}^{(j)})$;
4     for $i = 1$ to $n_j$ do // easy samples first
5         if $L(y_i^{(j)}, f(\boldsymbol{x}_i^{(j)}, \mathbf{w})) < \lambda + \gamma \frac{1}{\sqrt{i}+\sqrt{i-1}}$ then $v_i^{(j)} = 1$ ; // select this sample
6         else $v_i^{(j)} = 0$; // not select this sample
7     end
8 end
9 return v
```

As shown, Algorithm 3 selects samples in terms of both the easiness and the diversity. Specifically:

- Samples with $L(y_i, f(\mathbf{x}_i, \mathbf{w})) < \lambda$ will be selected in training $(v_i = 1)$ in Step 5. These samples represent the "easy" examples with small losses.

- Samples with $L(y_i, f(\mathbf{x}_i, \mathbf{w})) > \lambda + \gamma$ will not be selected in training $(v_i = 0)$ in Step 6. These samples represent the "complex" examples with larger losses.

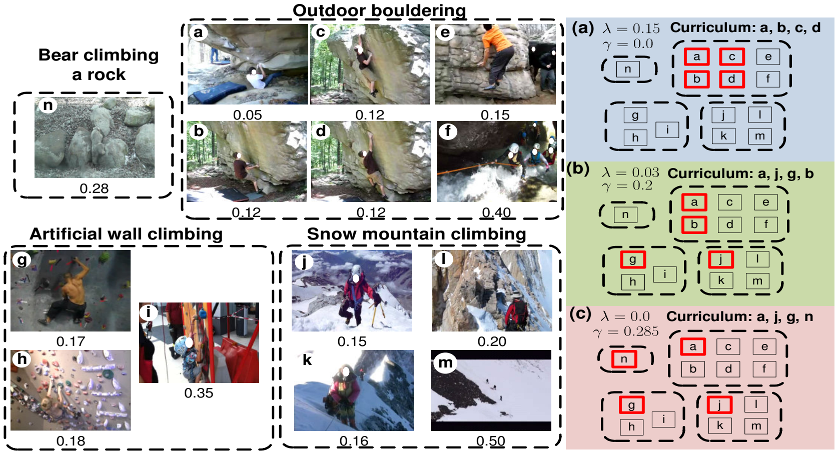

Outdoor bouldering
Bear climbing a rock
n
0.28
a
0.05
c
0.12
e
0.15
b
0.12
d
0.12
f
0.40
Artificial wall climbing
g
0.17
i
0.35
h
0.18
Snow mountain climbing
j
0.15
l
0.20
k
0.16
m
0.50
(a) $\lambda = 0.15$ $\gamma = 0.0$ Curriculum: a, b, c, d
(b) $\lambda = 0.03$ $\gamma = 0.2$ Curriculum: a, j, g, b
(c) $\lambda = 0.0$ $\gamma = 0.285$ Curriculum: a, j, g, n

FIGURE 7.2: An example on samples selected by Algorithm 3. A colored block denotes a curriculum with given λ and γ, and the bold (red) box indicates the easy sample selected by Algorithm 3.

- Other samples will be selected by comparing their losses to a threshold $\lambda + \frac{\gamma}{\sqrt{i}+\sqrt{i-1}}$, where i is the sample's rank w.r.t. its loss value within its group. The sample with a smaller loss than the threshold will be selected in training. Since the threshold decreases considerably as the rank i grows, Step 5 penalizes samples monotonously selected from the same group.

Example 7.2. *We study a tractable example that allows for clearer diagnosis in Fig. 7.2, where each keyframe represents a video sample on the event "Rock Climbing" of the TRECVID MED data [5], and the number below indicates its loss. The samples are clustered into four groups based on the visual similarity. A colored block on the right shows a curriculum selected by Algorithm 3. When $\gamma = 0$, as shown in Fig. 7.2(a), SPLD, which is identical to SPL, selects only easy samples (with the smallest losses) from a single cluster. Its curriculum thus includes duplicate samples like b, c, d with the same loss value. When $\lambda \neq 0$ and $\gamma \neq 0$ in Fig. 7.2(b), SPLD balances the easiness and the diversity, and produces a reasonable and diverse curriculum: a, j, g, b. Note that even if there exist 3 duplicate samples b, c, d, SPLD only selects one of them due to the decreasing threshold in Step 5 of Algorithm 3. Likewise, samples e and j share the same loss, but only j is selected as it is better in increasing the diversity. In an extreme case where $\lambda = 0$ and $\gamma \neq 0$, as illustrated in Fig. 7.2(c), SPLD selects only diverse samples, and thus may choose outliers, such as the sample n which is a confusable video about a bear climbing a rock. Therefore, considering both easiness and diversity seems to be more reasonable than considering either one alone. Physically the parameters λ and γ together correspond to the "age" of the model, where λ focuses on easiness whereas γ stresses diversity.*

7.4.4 Self-paced function with dropout

In many problems, only weak annotations are available. For example, the videos might be weakly labeled by the surrounding metadata such as title or descriptions. We call these samples are weakly labeled. The weakly-labeled videos can be collected without any manual effort, and its amount is thus orders of magnitude larger than that of any manually-labeled video collection. Unlike the manual labels, the weak labels are noisy and have both low accuracy and low recall: the weakly labeled concepts may not present in the video content and concepts not in the web label may appear in the video.

The labels in big weakly labeled data are much noisier than manually labeled data, and as a result, we found that the learning is prone to overfitting the noisy labels. To address this issue, inspired by the dropout technique in deep learning [152], we propose a dropout strategy for weakly labeled learning. It is implemented in the self-paced function. Define

$$r_i(p) \sim \text{Bernoulli}(p) + \epsilon, (0 < \epsilon \ll 1), \tag{7.10}$$

where $\mathbf{r}$ is a column vector of independent Bernoulli random variables with the probability p of being 1. Each of the element equals the addition of r_i and a small positive constant ϵ.

Then we can define the self-paced functions with dropout. For example, the binary self-paced function with dropout becomes:

$$f(\mathbf{v}; \lambda, p) = -\lambda \|\mathbf{r} \cdot \mathbf{v}\|_1, \tag{7.11}$$

and the linear self-paced function with dropout becomes:

$$f_l(\mathbf{v}; \lambda, p) = \frac{1}{2}\lambda \sum_{i=1}^{n} (\frac{1}{r_i} v_i^2 - 2v_i). \tag{7.12}$$

Denote $\mathbb{E}_{\mathbf{w}} = \sum_{i=1}^{n} v_i \ell_i + f(\mathbf{v}; \lambda)$ as the objective with the fixed model parameters $\mathbf{w}$ without any constraint, and the optimal solution $\mathbf{v}^* = [v_1^*, \cdots, v_n^*]^T = \arg\min_{\mathbf{v} \in [0,1]^n} \mathbb{E}_{\mathbf{w}}$. For binary self-paced function in Eq. (7.11), we have:

$$\mathbb{E}_{\mathbf{w}} = \sum_{i=1}^{n} (\ell_i - r_i \lambda) v_i \tag{7.13}$$

The closed-form solution is:

$$v_i^* = \begin{cases} 1 & \ell_i < r_i \lambda \\ 0 & \ell_i \geq r_i \lambda \end{cases} \tag{7.14}$$

For linear self-paced function in Eq. (7.12), the closed-form solution is:

$$\mathbb{E}_{\mathbf{w}} = \sum_{i=1}^{n} \ell_i v_i + \lambda(\frac{1}{2r_i} v_i^2 - v_i); \tag{7.15}$$

$$\frac{\partial \mathbb{E}_{\mathbf{w}}}{\partial v_i} = \ell_i + \lambda v_i / r_i - \lambda = 0; \Rightarrow v_i^* = \begin{cases} r_i(-\frac{1}{\lambda}\ell_i + 1) & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda \end{cases}. \tag{7.16}$$

The dropout effect can be demonstrated in the closed-form solutions in Eq. (7.14) and Eq. (7.16): with the probability $1 - p$, v_i^* approaches 0; with the probability p, v_i^* approaches the solution of the plain regularizer discussed in Eq. (7.4) and Eq. (7.5). Recall the self-paced function defines a scheme for learning, and the self-paced function with dropout represent new learning schemes.

When the base learner is neural networks, the proposed dropout can be used combined with the classical dropout in [152]. The term dropout in this paper refers to dropping out samples in the iterative learning. By dropping out a sample, we drop out its update to the model. It is useful for noisy data. When samples with incorrect noisy labels update a model, it will encourage the model to select more noisy labels. The dropout strategy prevents overfitting to noisy labels. Experimental results substantiate this argument. In practice, we recommend setting two Bernoulli parameters for positive and negative samples on imbalanced data. Empirically, we apply a much smaller probability p on the negative samples than on the positive samples. In other words, we encourage models to mainly drop out negative samples.

7.5 Discussions

7.5.1 Theoretical Justification

Interestingly, it turns out that Algorithm 2 actually optimizes an underlying non-convex robust loss on the noisy data. To show this, let $v^*(\lambda, \ell)$ represent the optimal weight of v for a loss term ℓ imposed on a training sample in Eq (7.3), where

$$v^*(\lambda, \ell) = \text{argmin}_{v \in [0,1]} \quad v\ell + f(v, \lambda). \tag{7.17}$$

For convenience of notation, let the curriculum region be the full space. According to [153], the latent objective has the form of $\mathbb{E}_\ell = \sum_{i=1}^{n} F_\lambda(\ell_i)(\lambda > 0)$ with a latent loss function $F_\lambda(\ell)$ obtained by integrating the loss variable from $v^*(\lambda, \ell)$, i.e.,

$$F_\lambda(\ell) = \int_0^\ell v^*(\lambda; l) dl. \tag{7.18}$$

Note that in the above ℓ and l means loss variables in the latent loss function $F_\lambda(\ell)$ and the optimal weight function $v^*(\lambda, l)$, whereas ℓ_i denotes the loss value actually calculated on the i-th sample. Incorporating the binary and linear self-paced functions in Eq. (7.18), the latent objective becomes:

$$F_\lambda^b(\ell) = \min(\ell, \lambda) \tag{7.19}$$

$$F_\lambda^l(\ell) = \mathbf{I}(\ell \geq \lambda)\frac{\lambda}{2} + \mathbf{I}(\ell < \lambda)(\ell - \frac{\ell^2}{2\lambda}) \tag{7.20}$$

Eq. (7.19) and Eq. (7.20) are two common non-convex regularized penalties in the machine learning community, where Eq. (7.19) is the Capped-Norm based Penalty(CNP) [154, 155] and Eq. (7.20) is the Minimax Convex Plus (MCP) [156]. It has been showed that both CNP and MCP can be used as robust loss functions that threshold the samples of greater loss [157]. Therefore, Algorithm 2 actually minimizes a non-convex robust loss derived from the original loss in the base learner (e.g. hinge loss). On clean data, the effect of the robust loss may not be evident, but on noisy data, without the robust loss, the model can be easily dominated by a few noisy samples or outliers. Experimental results substantiate this argument, where we observed that the robust loss leads to more accurate results than the original loss on the weakly labeled data.

The proposed theory can be theoretically justified from two independent perspectives. From the learning perspective, it mimics the human and animal learning process that learns a model gradually from confident to less confident examples in the noisy data. From the optimization perspective, it minimizes a non-convex robust loss (CNP or MCP) on the noisy data. The robust loss tends to depress samples with noisy labels or outliers. Due to the nature of non-convexity, it utilizes the curriculum and self-paced learning, which have been demonstrated to be instrumental in avoiding bad local minima in non-convex problems [119, 120]. Interestingly, Meng et al. [153] proved that when λ is fixed, Algorithm 2, in fact, is identical to the Majorization-Minimization algorithm [158], a popular solver for non-convex problems [153]. Based on the understanding, one can justify the role of the curriculum region, i.e. the curriculum confines the search space of a non-convex problem to some reasonable subspace which tends to improve the quality of the starting value and the final solution. The dropout methods on the other hand, prevent overfitting in the non-convex optimization problem.

7.5.2 Limitations

Since the publication of SPCL in 2015, it has helped inspire a number of studies in a variety of fields. For example, in the noisy data learning [159–161], information retrieval [162], multimedia retrieval [163, 164], visual saliency detection [165], matrix factorization [166], multi-task/objective learning [167, 168], natural language processing [169], deep learning [170, 171], machine learning theory [172], and so on.

However, we did observe a number of limitations of the current SPCL model. First, the fundamental learning philosophy of SPL/CL/SPCL is 1) learning needs to be conducted iteratively using samples organized in a meaningful sequence; 2) models are becoming more complex in each iteration. However, the learning philosophy may not applicable to every learning problem. For example, in many problems where training data, especially small training data, are carefully selected and the spectrum of learning difficulty of training samples is controlled. We found the proposed theory may not outperform the conventional training methods. Second, the performance of SPCL can be unstable to the random starting values. This phenomenon can be intuitively explained in the context of education, it is impossible for students to predetermine what to learn before they actually learning anything. To address this, the curriculum needs to be meaningful so that it can provide some supervision in the first few iterations. However, precisely deriving curriculum region from prior knowledge seems to be an open question. When the prior knowledge is unavailable, a common approach is to first train a model on 50%, randomly selected data, and use the model to initialize the samples and their weights in the first iteration. Third, the age parameters λ are very important hyper-parameters to tune. In order to tune the parameters, the proposed theory requires a labeled validation set that follows the same underlying distribution of the test set. Intuitively, it is analogous to the mock exam whose purposes are to let students realize how well they would perform on the real test data, and more importantly have a better idea of what to study.

In implementation, we found some engineering tricks to apply the theory to real-world problems. First, the parameters λ (and γ in the diversity learning scheme) should be tuned by the statistics collected from the ranked samples, as opposed to the absolute values. For example, instead of setting λ to an absolute value, we rank samples by their loss in increasing order, then set λ as the loss of the top nth sample. As a result, the top $n-1$ samples will be used in training. The nth sample will have 0 weights and will not be used in training, and so does the samples ranked after it. In the next iteration we may increase λ to be the loss of the top 1.5n sample. This strategy avoids selecting too many or too few samples at a single iteration and seems to be robust. Second, for unbalanced datasets, two sets of parameter λ were introduced: λ_+ for positive and λ_- for negative samples in order to pace positive and negative separately. This trick lead to a balance training data set in each iteration. Third, for the convex loss function L in the off-the-shelf model, if we use the same training sets, we will end up with the same model, irrespective of iterative steps. In this case, at each iteration, we should test our model on the validation set, and determine when to terminate the training process. The converged model on a subset of training samples may perform better than the model trained on the whole training set. For example, Lapedriza et al. [173] found training detectors using a subset samples can yield better results. On noisy data, early stopping must be applied as training all samples is bound to introduce much noise in training. For non-convex loss function in the off-the-shelf model, the sequential steps affect the final model. Therefore, the early stopping may not be necessary on clean data sets.

7.6 Experiments using Diversity Scheme

We name SPCL with diversity learning scheme SPLD. We present experimental results on two tasks: event detection and action recognition. We demonstrate that our approach outperforms SPL on three real-world challenging datasets.

SPLD is compared against four baseline methods: 1) **RandomForest** is a robust bootstrap method that trains multiple decision trees using randomly selected samples and features [174]. 2) **AdaBoost** is a classical ensemble approach that combines the sequentially trained "base" classifiers in a weighted fashion [175]. Samples that are misclassified by one base classifier are given greater weight when used to train the next classifier in sequence. 3) **BatchTrain** represents a standard training approach in which a model is trained simultaneously using all samples; 4) **SPL** is a state-of-the-art method that trains models gradually from easy to more complex samples [120]. The baseline methods are a mixture of the well-known and the state-of-the-art methods on training models using sampled data.

7.6.1 Event Detection

Given a collection of videos, the goal of MED is to detect events of interest, e.g. "Birthday Party" and "Parade", solely based on the video content. The task is very challenging due to complex scenes, camera motion, occlusions, etc. The experiments are conducted on the largest collection on event detection: TRECVID MED13Test, which consists of about 32,000 Internet videos. There are a total of 3,490 videos from 20 complex events, and the rest are background videos. For each event 10 positive examples are given to train a detector, which is tested on about 25,000 videos. The official test split released by NIST (National Institute of Standards and Technology) is used. A Deep Convolutional Neural Network is trained on 1.2 million ImageNet challenge images from 1,000 classes [37] to represent each video as a 1,000-dimensional vector. Algorithm 3 is used.

By default, the group membership is generated by the spectral clustering, and the number of groups is set to 64. Following [142], LibLinear is used as the solver in Step 4 of Algorithm 3 due to its robust performance on this task. The performance is evaluated using MAP as recommended by NIST. The parameters of all methods are tuned on the same validation set.

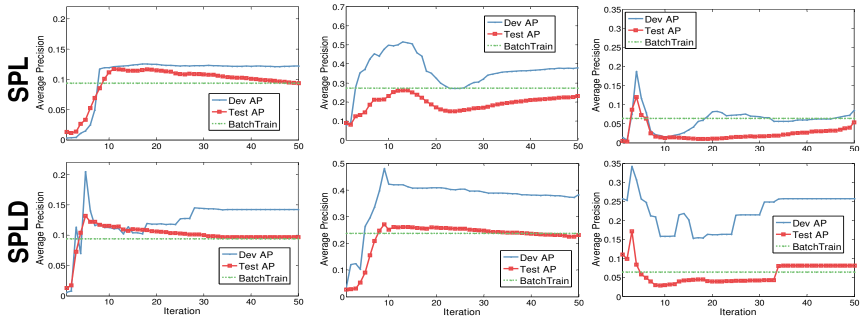
Table 7.2 lists the overall MAP comparison. To reduce the influence brought by initialization, we repeated experiments of SPL and SPLD 10 times with random starting values, and report the best run and the mean (with the 95% confidence interval) of the 10 runs. The proposed SPLD outperforms all baseline methods with statistically significant differences at the p-value level of 0.05, according to the paired t-test. It is worth emphasizing that MED is very challenging and 26% relative (2.5 absolute) improvement over SPL is a notable gain. SPLD outperforms other baselines on both the best run and the 10 runs average. RandomForest and AdaBoost yield poorer performance. This observation agrees with the study in literature [5] that SVM is more robust on event detection.

Table 7.2: MAP (x100) comparison with the baseline methods on MED.

Run Name	RandomForest	AdaBoost	BatchTrain	SPL	SPLD
Best Run	3.0	2.8	8.3	9.6	**12.1**
10 Runs Average	3.0	2.8	8.3	8.6±0.42	**9.8±0.45**

BatchTrain, SPL and SPLD are all performed using SVM. Regarding the best run, SPL boosts the MAP of the BatchTrain by a relative 15.6% (absolute 1.3%). SPLD yields another 26% (absolute 2.5%) over SPL. The MAP gain suggests that optimizing objectives with the diversity is inclined to attain a better solution. Fig. 7.3 plots the validation and test AP on three representative events. As illustrated, SPLD attains a better solution within fewer iterations than SPL, e.g. in Fig. 7.3(a) SPLD obtains the best test AP (0.14) by 6 iterations as opposed to AP (0.12) by 11 iterations in SPL. Studies have shown that SPL converges fast, while this observation further suggests that SPLD may lead to an even faster convergence. We hypothesize that it is because the diverse samples learned in the early iterations in SPLD tend to be more informative. The best Test APs of both SPL and SPLD are better than BatchTrain, which is consistent with the observation in [173] that removing some samples may be beneficial in training a better detector. As shown, Dev AP and Test AP share a similar pattern justifying the rationale for parameters tuning on the validation set.

Fig. 7.4 plots the curriculum generated by SPL and SPLD in a first few iterations on two representative events. As we see, SPL tends to select easy samples similar to what it has already learned, whereas SPLD selects samples that are both easy and diverse to the model. For example, for the event “E006 Birthday Party”, SPL keeps selecting indoor scenes due to the sample learned in the first place. However, the samples learned by



(a) E006: Birthday party (b) E008: Flash mob gathering (c) E023: Dog show

Figure 7.3: The validation and test AP in different iterations. Top row plots the SPL result and bottom shows the proposed SPLD result. The x-axis represents the iteration in training. The blue solid curve (Dev AP) denotes the AP on the validation set, the red one marked by squares (Test AP) denotes the AP on the test set, and the green dashed curve denotes the Test AP of BatchTrain which remains the same across iterations.

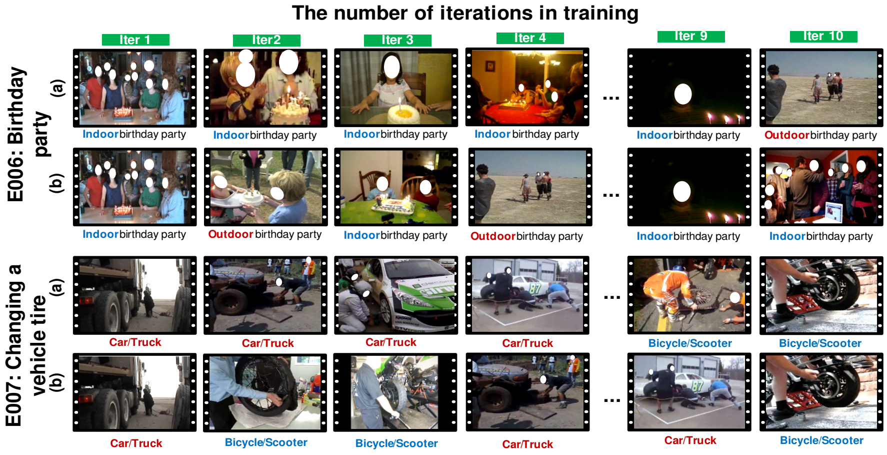


Figure 7.4: Comparison of positive samples used in each iteration by (a) SPL (b) SPLD.

SPLD are a mixture of indoor and outdoor birthday parties. For the complex samples, both methods leave them to the last iterations, e.g. the 10th video in “E007”.

7.6.2 Action Recognition

The goal is to recognize human actions in videos. Two representative datasets are used: Hollywood2 was collected from 69 different Hollywood movies [42]. It contains 1,707 videos belonging to 12 actions, splitting into a training set (823 videos) and a test set (884 videos). Olympic Sports consists of athletes practicing different sports collected from YouTube [41]. There are 16 sports actions from 783 clips. We use 649 for training and 134 for testing as recommended in [41]. The improved dense trajectory feature is extracted and further represented by the fisher vector [16, 176]. A similar setting discussed in Section 7.6.1 is applied, except that the groups are generated by K-means (K=128).

Table 7.3: MAP (x100) comparison with the baseline methods on Hollywood2 and Olympic Sports.

Run Name	RandomForest	AdaBoost	BatchTrain	SPL	SPLD
Hollywood2	28.20	41.14	58.16	63.72	**66.65**
Olympic Sports	63.32	69.25	90.61	90.83	**93.11**

Table 7.3 lists the MAP comparison on the two datasets. A similar pattern can be observed that SPLD outperforms SPL and other baseline methods with statistically significant differences. We then compare our MAP with the state-of-the-art MAP in Table 7.4. Indeed, this comparison may be less fair since the features are different in different methods. Nevertheless, with the help of SPLD, we are able to achieve the best MAP reported so far on both datasets. Note that the MAPs in Table 7.4 are obtained by recent and very competitive methods on action recognition. This improvement confirms the assumption that considering diversity in learning is instrumental.

Table 7.4: Comparison of SPLD to the state-of-the-art on Hollywood2 and Olympic Sports

Hollywood2		Olympic Sports	
Vig et al. 2012 [177]	59.4%	Brendel et al. 2011 [178]	73.7%
Jiang et al. 2012 [179]	59.5%	Jiang et al. 2012 [179]	80.6%
Jain et al. 2013 [40]	62.5%	Gaidon et al. 2012 [180]	82.7%
Wang et al. 2013 [16]	64.3%	Wang et al. 2013 [16]	91.2%
SPLD	**66.7%**	**SPLD**	**93.1%**

7.7 Experiments on Noisy Data

This section verifies the accuracy and the scalability of the proposed method on learning concept detectors on weakly labeled video data. In each dataset, the labels are automatically derived from the textual metadata without manual supervision. The experiments are conducted on two public benchmarks, where FCVID is by far one of the biggest manually annotated video set, and the YFCC100M is the largest multimedia benchmark.

7.7.1 Experimental Setup

Dataset and Feature Fudan-Columbia Video Dataset (FCVID) contains 91,223 YouTube videos (4,232 hours) from 239 categories. It covers a wide range of concepts like activities, objects, scenes, sports, etc. [181]. Each video is manually labeled to one or more categories. In our experiments, we do not use the manual labels in training, but instead we automatically generate the web labels according to the concept name appearance in the video metadata. The manual labels are used only in testing to evaluate our and baseline methods. Following [181], the standard train/test split and the same static Convolution Neural Networks(CNN) feature from [181] are used to have a fair comparison to existing methods. The second set YFCC100M [59] contains about 800,000 videos on Flickr with metadata such as the title, tags, the uploader, etc. There are no manual labels on this set and we automatically generate the web labels from the metadata. We use the features provided in [11] where we first extract the keyframe level the VGG neural network features [182] and create a video feature by average pooling. The same features are used across different methods on each dataset. Since there are no annotations, we train the concept detectors on the most 101 frequent latent topics in the video metadata. On YFCC, we use a data-driven approach to determine the concept vocabulary. Given a collection of video with metadata, we run topic models to surface frequent objects, scenes, actions or events. The list of automatically generated latent topics is manually examined to obtain the final concept vocabulary. In this way, we can ensure every concept has a reasonable number of training samples.

Baselines The proposed method is compared against the following five baseline methods which cover both the classical and the recent representative learning algorithms on weakly-labeled data. *BatchTrain* trains a single SVM model using all samples with noisy labels. *AdaBoost* is a classical ensemble approach that combines the sequentially trained "base" classifiers in a weighted fashion [175]. *Self-Paced Learning (SPL)* is a classical method where the curriculum is generated by the learner itself [120]. *BabyLearning* is a recent method that simulates baby learning by starting with few training samples and fine-tuning using more weakly labeled videos crawled from the search engine [151]. We build a search engine that indexes the textual metadata and retrieves videos using concept words based on Lucene [82]. *GoogleHNM* uses the hard negative mining strategy in [48]. On FCVID, we use the YouTube topic API to acquire the noisy labels whereas on YFCC100M we obtain the noisy labels by the Lucene search engine.

Evaluation Metrics On FCVID, as the manual labels are available, the performance is evaluated in terms of the precision of the top 5 and 10 ranked videos (P@5 and P@10) and mean Average Precision (mAP) of 239 concepts. On YFCC100M, since there are no manual labels, for evaluation, we apply the detectors to a third public video collection called TRECVID MED which includes 32,000 Internet videos [5]. We apply the detectors trained on YFCC100M to the TRECVID videos and manually annotate the top 10 detected videos of each method for the 101 concepts.

Our Model We build our method on top of a pre-trained convolutional neural network as the low-level features, i.e. static CNN features on FCVID and VGG-16 features [183] on YFCC100M. The concept detectors are trained based on a hinge loss cost function. Algorithm 2 is used to train the concept models iteratively and the λ stops increasing after 100 iterations. We automatically generate noisy web labels based on the video metadata. For the videos with noisy positive labels, we group them based on their latent topics, and derive a partial-order curriculum in Definition 7.3. The hyper-parameters of all methods including the baseline methods are tuned on the same validation set. On FCVID, the set is a small training subset with manual labels whereas on YFCC100M it is a proportion of noisy training set.

7.7.2 Experiments on FCVID

Table 7.5 compares the precision and mAP of different methods where the best results are highlighted. As we see, the proposed SPCL with dropout significantly outperforms all baseline methods, with a significant difference at *p*-level of 0.05. For example, it outperforms the best baseline on 194 out of 239 concepts. The promising experimental results substantiate our theoretical analysis in Chapter 7.5. With the proposed model, the binary and linear regularizer yield a similar accuracy on this dataset. The performance difference between SPCL with and without dropout demonstrates the efficacy of the proposed dropout technique, and the difference between SPL and SPCL indicates the benefit of incorporating prior knowledge as the partial-order curriculum.

Note that SPCL does not use any manual labels in training, but interestingly, its accuracy is comparable with the model trained on 35,850 videos with ground truth labels in [181]. To investigate the potential of training concepts on the weakly labeled data setting, we apply SPCL on the data subsets of different sizes. Specifically, we randomly split the FCVID training set into the subset of 200, 500, 1,000, and 2,000 hours of videos, and train the models on each subset. The models are then tested on the same standard test set. Table 7.6 lists the results. As we see, the accuracy of SPCL on weakly labeled data increases along with the growth of the size of noisy data. The accuracy on 2,000 hours of videos with noisy web labels turns out to be better than the model trained on 500 hour of manually labeled data. Recall FCVID is one of the biggest manually annotated set which contains about 2,000 hours of annotated videos. According to the results, we hypothesize that with more weakly labeled data, which is not hard to obtain, our method can potentially outperform models trained on any existing manually-labeled data.

Table 7.5: Performance comparison on FCVID.

Method	P@5	P@10	mAP
BatchTrain	0.782	0.763	0.469
Adaboost	0.456	0.412	0.293
SPL	0.793	0.754	0.414
GoogleHNM	0.781	0.757	0.472
BabyLearning	0.834	0.817	0.496
SPCL w/o dropout (binary)	0.857	0.843	0.521
SPCL w/. dropout (linear)	**0.893**	**0.877**	**0.566**
SPCL w/. dropout (binary)	**0.893**	**0.878**	**0.567**

Table 7.6: MAP comparison of models trained using weak labels and ground-truth labels on different subsets of FCVID. Noted that SPCL is trained using noisy web labels while $*$ is trained using ground truth labels

Dataset Size	200h	500h	1000h	2000h
SPCL	0.413	0.480	0.520	0.567
BatchTrain*	0.485	0.561	0.604	0.638

7.7.3 Experiments on YFCC100M

Since there are no manual labels on YFCC100M, to evaluate the performance, we manually annotate the top 10 videos in the test set and report their precisions in Table 7.7. A similar pattern can be observed where the comparisons substantiate the rationality of the proposed partial-order curriculum and the dropout technique. The promising results on the largest multimedia set YFCC100M verify the scalability of the proposed method.

Table 7.7: Performance comparison on YFCC100M.

Method	P@3	P@5	P@10
BatchTrain	0.535	0.513	0.487
Adaboost	0.341	0.327	0.282
SPL	0.485	0.463	0.454
GoogleHNM	0.541	0.525	0.500
BabyLearning	0.548	0.519	0.466
SPCL binary w/o dropout (binary)	0.607	0.608	0.589
SPCL w/. dropout (linear)	**0.667**	**0.663**	**0.649**
SPCL w/. dropout (binary)	**0.660**	**0.640**	**0.625**

7.8 Summary

We proposed a novel learning regime called self-paced curriculum learning (SPCL), which imitates the learning regime of humans/animals that gradually involves from easy to more complex training samples into the learning process. The component of SPCL is of physical interpretation. The off-the-shelf models, such as SVMs, deep neural networks, and regression models, correspond to students. The self-paced functions correspond to learning schemes used by students to solve specific problems. The curriculum region corresponds to the prior knowledge provided from an oracle or an instructor so that learning can be processed in a desired direction. The proposed SPCL can exploit both prior knowledge before training and dynamical information extracted during training. The novel regime is analogous to an "instructor-student-collaborative" learning mode, as opposed to "instructor-driven" in curriculum learning or "student-driven" in self-paced learning. We presented compelling understandings for curriculum learning and self-paced learning, and revealed that the underlying robust loss function the model tries to optimize. We discussed several concrete implementations in the proposed SPCL framework, and presented experiments on a number of datasets to demonstrate its promising results in learning concept detectors.

SPCL is a general learning framework. Despite the promising results, we did observe a number of limitations as discussed Section 7.5.2. This thesis is not able to completely address all limitations. We believe the proposed theory is still immature, and thus needs further research.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this thesis, we studied a fundamental research problem of searching and learning semantic information in video content at a very large scale. We proposed several novel methods focusing on improving accuracy, efficiency and scalability. The proposed method demonstrated promising results on web-scale video learning, search and understanding. Based on the proposed methods, we implement E-Lamp Lite, the first of its kind large-scale semantic search engine for Internet videos. The extensive experiments demonstrated that the methods are able to surpass state-of-the-art accuracy on multiple datasets. In addition, our method can efficiently scale up the search to hundreds of millions videos, and only takes about 0.2 second to search a semantic query on a collection of 100 million videos, 1 second to process a hybrid query over 1 million videos. There are two research issues to be addressed. To the best of our knowledge, E-Lamp Lite is also the first content-based video retrieval system that is capable of indexing and searching a collection of 100 million videos.

There are a number of future directions to improve the current system. For example, the current concept detection mainly focuses on video-level recognition and on recognizing coarse-grained nouns and verbs. It can neither understand the relation between the entities, nor detect subtle or small objects, actions or events in the video. A promising is to apply advanced computer vision techniques to address this limitation.

8.2 Application: Visual Memory QA

In this section, we demonstrate an interesting application that is built on the proposed E-Lamp Lite system. Our point is to show that the proposed system can serves as an important fundamental engine for intelligent systems. The demo video can be found at https://www.youtube.com/watch?v=wr19v1GofAs

The prevailing of mobile devices and cloud services has led to an unprecedented growth of personal photo and video data. A recent study shows that the queries over personal photos or videos are usually task- or question-driven [4]. For question-driven queries, users seem to be using photos or videos as a mean to recover pieces from their own memories, i.e. looking for a specific name, place or date. For example, a user might ask "what was the last time we went hiking?"; "did we have pizza last week?" or "with whom did I have dinner in AAAI 2015?".

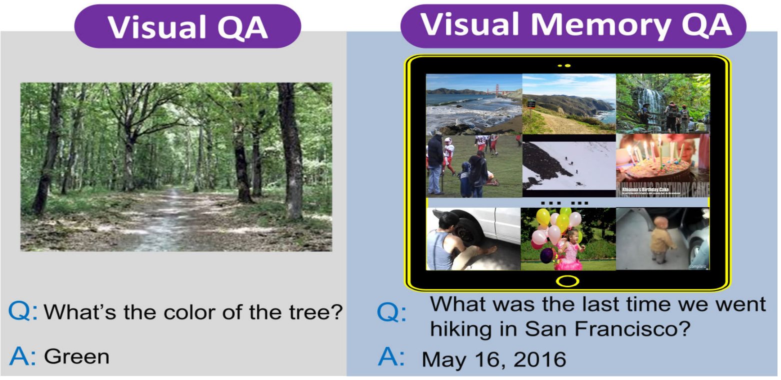

Visual QA
Visual Memory QA
Q: What's the color of the tree?
A: Green
Q: What was the last time we went hiking in San Francisco?
A: May 16, 2016

FIGURE 8.1: Comparison of Visual QA & Visual Memory QA.

We define the problem of seeking answers about the user's daily life discovered in his or her personal photo and video collection as MemexQA (Visual Memory Question Answering). As about 80% of personal photos and videos do not have metadata such as tags or titles [4], this functionality can be very useful in helping users find information in their personal photos and videos. Visual Memory QA is a novel problem and has two key differences from VQA (Visual QA) [184]: first the user is able to ask questions over a collection of photos or videos in Visual Memory QA as opposed to a single image in VQA. As shown in Fig. 8.1, given an image it is trivial for an adult to answer a question in VQA. However, it is considerably more difficult for the same adult to answer questions in MemexQA. This is particularly difficult and time consuming to answer questions over a collection videos. Second, the question space in MemexQA is a subset of that in VQA, which only includes the questions a user might ask later to recall his or her memories. Because of the two differences, Visual Memory QA is expected to be more useful in practice.

To address this novel problem, we introduce a prototype system that is built on the proposed E-Lamp Lite system. The prototype can automatically analyzes the content of personal videos/photos without user-generated metadata, and offers a conversational interface to answer questions discovered from the user's personal videos/photos. Technically, it can be regarded as an end-to-end neural network, consisting of three major components: a recurrent neural network to understand the user question, a content-based video engine to analyze and find relevant videos, and a multi-channel attention neural network to extract the answer. To the best of our knowledge, the proposed system is the first to answer personal questions discovered in personal photos or videos.

As shown in Fig. 8.2, the proposed model is inspired by the classical text QA model [185], consisting of three major components: a recurrent neural network to understand the user question, a content-based video engine to find the relevant videos, and a multi-channel attention feed-forward neural network to extract the answer. Each component is pre-trained on its own task, and then the first and the third components are fine-tuned on our annotated benchmark data by Back Propagation.

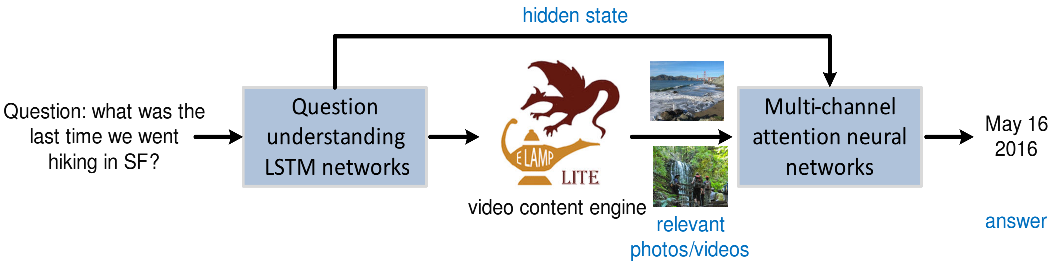

hidden state
Question: what was the last time we went hiking in SF?
Question understanding LSTM networks
E LAMP LITE
video content engine
relevant photos/videos
Multi-channel attention neural networks
May 16 2016
answer

FIGURE 8.2: Framework of the proposed Visual Memory QA system.

In the recurrent neural network, the task is to understand the question and classify it into a predefined answer type. We predefine a set of question and answer types based on their frequencies in Flickr visual search logs [4]. See Table 8.1. A two-layer LSTM neural network is incorporated as the classifier where the embedding of each word in the question is sequentially fed into the LSTM units. As the answer types are mutually exclusive, a softmax logistic loss is employed to train the network. Besides, this question understanding component is also responsible for parsing the question to extract the named entity (person, organization, place and time).

TABLE 8.1: Question and answer types in the proposed system.

Question Type	Answer Type	Example
which	photo, video	show me the photo of my dog?
when	date, year, season, hour, etc.	What was the last time we went hiking?
where	scene, gps, city, country, etc.	Where was my brother's graduation ceremony in 2013?
what	action, object, activity, etc.	What did we play during this spring break?
who	name, face, etc.	With whom did I have dinner in AAAI 2015?
how many	number	How many times have I had sushi last month?
yes/no	yes, no	Did I do yoga yesterday?

The second component is the proposed content video/photo engine (E-Lamp Lite) that can automatically understand and index personal videos purely based on the video content. It takes a natural language sentence as the input, and outputs a list of semantically relevant videos, i.e. text-to-video [58]. The top ranked relevant videos are fed into the third component.

The last component is a neural network to extract the answer. It receives, from the question understanding network, a hidden state that embeds the information about the predicted answer type, and the top ranked relevant videos from the video content engine. Each relevant video is associated with information organized into channels, such as the timestamp, the action concepts, scene concepts, object concepts and, in some cases, the GPS coordinates. The task now switches to localizing the answer in the multiple input channels. For example, the attention should be on timestamp for “when” questions, and on food concepts for “what did we eat” questions. This is now achieved by a multi-channel attention feed-forward neural network. For the current prototype, a few manual templates are also employed to further improve the accuracy.

We presented a novel and promising Visual Memory QA system, an intelligent agent or chatbot that can answer questions about users’ daily lives discovered in their personal photos and videos. We have developed a prototype system that can efficiently answer questions over 1 million personal videos. We will release an open benchmark dataset on this task in future.

Appendix A

Proof

***Theorem* 3.4*: the thresholding and the top-k thresholding results are optimal solutions of Eq. (3.1) in special cases.*

Proof. Suppose $\mathbf{v} \in [0,1]^m$ represents the adjusted m-dimensional representation, and $\mathbf{d} \in [0,1]^m$ represents the vector $f_p(\mathbf{D})$. Define the regularization term $g(\mathbf{v};\alpha,\beta)$ as:

$$g(\mathbf{v};\alpha,\beta) = \frac{1}{2}\beta^2\|\mathbf{v}\|_0,$$

and denote the objective function value of Eq. (3.1) in the paper as $\mathbb{E}(\mathbf{v},\beta)$. Then the optimization problem without constraint is reformulated as:

$$\begin{aligned}\min_{\mathbf{v}} \mathbb{E}(\mathbf{v},\beta) &= \frac{1}{2}\|\mathbf{v}-\mathbf{d}\|_2^2 + \frac{1}{2}\beta^2\|\mathbf{v}\|_0 \\ &= \frac{1}{2}\sum_{i=1}^{m}\left((v_i-d_i)^2 + \beta^2|v_i|_0\right).\end{aligned}$$

It is easy to see that this optimization can be decomposed into m sub-optimization problems for $i = 1, 2, \cdots, m$ as

$$\min_{\mathbf{v}} \mathbb{E}(v_i,\beta) = \frac{1}{2}(v_i-d_i)^2 + \frac{1}{2}\beta^2|v_i|_0. \tag{A.1}$$

For any $v_i \neq 0$ it holds that $|v_i|_0 = 1$ in this case. Thus the minimum of Eq. (A.1) is obtained at the minimal value of the first term, where $v_i^* = d_i$, and the corresponding optimal objective value is $\mathbb{E}(v_i^*,\beta) = \frac{1}{2}\beta^2$.

For $v_i = 0$, we have that $\mathbb{E}(v_i,\beta) = \frac{1}{2}d_i^2$.

It is then easy to deduce that the optimum of Eq. (A.1) can be calculated at:

$$v_i^* = \begin{cases} d_i, & d_i \geq \beta \\ 0, & \text{otherwise}, \end{cases}$$

and thus the solution of the original problem is

$$\mathbf{v}^* = [v_1^*, v_2^*, \cdots, v_m^*]^T.$$

Denote the kth largest component of $\mathbf{d}$ is $\widetilde{d}_k$. Then if we want to get the k-sparse solution of Eq. (3.1), we just need to set that $\widetilde{d}_k \leq \beta < \widetilde{d}_{k+1}$. Then the solution of the problem keeps top-k largest elements of $\mathbf{d}$ while thresholds others to 0. This corresponds the thresholding and the top-k thresholding results. The proof is completed. □

***Theorem 3.3**: the optimal solutions of Eq. (3.1) (before or after normalization) is logically consistent with its given HEX graph.*

Proof. Suppose $\mathbf{v} \in [0,1]^m$ represents the adjusted m-dimensional representation, and $G = (N, E_h, E_e)$ represents the given HEX graph. For any concept $(n_i, n_j) \in E_h$, according to Algorithm 1, we have $v_i \geq v_j$. Therefore $\forall n_k, n_j \in V$, $n_k \in \alpha(n_j)$, we have $v_k \geq v_j$, where $\alpha(n_j)$ is a set of ancestor of n_j in G_h . This means condition 1 in Definition 3.2 is satisfied.

Suppose $\exists n_p \in \bar{\alpha}(n_i)$, $\exists n_q \in \bar{\alpha}(n_j)$ s.t. $(n_p, n_q) \in E_e$. According to Algorithm 1, the constraints ensure that if $(n_p, n_q) \in E_h$, $v_p v_q = 0$, which breaks down into three cases: 1) $v_p \neq 0, v_q = 0$, 2) $v_p = 0, v_q \neq 0$, and 3) $v_p = v_q = 0$. According to the condition 1 in Definition 3.2, we have $v_p \geq v_i$ and $v_q \geq v_j$ so for case 1) we have $v_j \leq v_q = 0$; for case 2) $v_i \leq v_p = 0$; for case 3) $v_i = 0$ and $v_j = 0$. Therefore in all cases, either v_i or v_j is nonzero. When $i = p$ and $j = q$, it trivially holds that $v_p v_q = v_i v_j = 0$. This means condition 2 in Definition 3.2 is satisfied.

According to Definition 3.2, $\mathbf{v}$ satisfies the two conditions and thus is logically consistent.

The normalization method discussed in the paper does not change nonzero scores to zero or vice versa. If $\mathbf{v}$ is consistent with the exclusion relation in the given HEX graph (condition 2), then after normalization it is still consistent. The normalization method multiples a constant factor to each dimension of $\mathbf{v}$ so it would not change the ranking order of the concepts. Therefore the normalization also preserves the hierarchical relation (condition 1). The proof is completed. □

***Lemma 6.1**: for the self-paced functions in Section 6.4.2, the proposed method finds the optimal solution for Eq. (6.13).*

Proof. Consider the objective of Eq. (6.13). Suppose $\mathbf{y}^* = [y_1^*, ..., y_n^*]^T$ is a solution found by the gradient descent method in Section 6.4.2. According to Eq. (6.14), $\forall y_i \in \{-1, +1\}$ and $\forall v_i \in [0, 1]$, we have:

$$\mathbb{E}_\Theta(y_i^*, v_i; k) \leq \mathbb{E}_\Theta(y_i, v_i; k). \tag{A.2}$$

Therefore $\forall \mathbf{y}$, $\forall \mathbf{v}$, the following inequations hold:

$$\mathbb{E}_\Theta(\mathbf{y}^*, \mathbf{v}; k) = \sum_{i=1}^{n} \mathbb{E}_\Theta(y_i^*, v_i; k) \leq \sum_{i=1}^{n} \mathbb{E}_\Theta(y_i, v_i; k) = \mathbb{E}_\Theta(\mathbf{y}, \mathbf{v}; k). \tag{A.3}$$

In other words, $\mathbf{y}^*$ found by Eq. (6.14) is the global optimum for Eq. (6.13). Now consider the objective with the fixed $\mathbf{y}^*$. The functions in Section 6.4.2, $f(\mathbf{v})$ are convex functions of $\mathbf{v}$, so Eq. (6.13) is a convex function of $\mathbf{v}$. Suppose that $\mathbf{v}^*$ is a solution found by gradient descent, due to the convexity, $\mathbf{v}^*$ is the global optimum for Eq. (6.13). Therefore, $\mathbf{y}^*, \mathbf{v}^*$ is the global optimal solution for Eq. (6.13). □

Theorem 6.2*: the algorithm in Fig. 6.2 converges to a stationary solution for any fixed C and k.*

Proof. Let the superscript index the variable value in that iteration, e.g. $\mathbf{v}^{(t)}$ represents the value of $\mathbf{v}$ in the t^{th} iteration. Denote $\Theta^{(t)} = \Theta_1^{(t)}, ..., \Theta_m^{(t)}$. $\mathbf{y}^{(0)}$ and $\mathbf{v}^{(0)}$ are arbitrary initial values in their feasible regions. As Eq. (6.12) is a quadratic programming problem, the solution $\Theta^{(t)}$ is the global optimum for $\mathbb{E}_{\mathbf{y},\mathbf{v}}$, i.e.

$$\mathbb{E}(\Theta^{(t)}, \mathbf{y}^{(t-1)}, \mathbf{v}^{(t-1)}) \leq \mathbb{E}(\Theta^{(t-1)}, \mathbf{y}^{(t-1)}, \mathbf{v}^{(t-1)}). \tag{A.4}$$

According to Lemma 6.1, $\mathbf{v}, \mathbf{y}$ are also global optimum for $\mathbb{E}_\Theta$, i.e.

$$\mathbb{E}(\Theta^{(t)}, \mathbf{y}^{(t)}, \mathbf{v}^{(t)}) \leq \mathbb{E}(\Theta^{(t)}, \mathbf{y}^{(t-1)}, \mathbf{v}^{(t-1)}). \tag{A.5}$$

Substitute Eq. (A.5) back into Eq. (A.4), we have that $\forall t \geq 1$,

$$\mathbb{E}(\Theta^{(t)}, \mathbf{y}^{(t)}, \mathbf{v}^{(t)}) \leq \mathbb{E}(\Theta^{(t-1)}, \mathbf{y}^{(t-1)}, \mathbf{v}^{(t-1)}). \tag{A.6}$$

Eq. (A.6) indicates that the objective decreases in every iteration. Since the objective $\mathbb{E}$ is the sum of finite elements, it is bounded from below. Consequently, according to [186], it is guaranteed that Alg. 6.2 (an instance of CCM algorithm) converges to a stationary solution of the problem. □

Theorem 7.5: For training samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, given a curriculum γ defined on it, the feasible region, defined by,

$$\Psi = \{\mathbf{v} | \mathbf{a}^T \mathbf{v} \leq c\} \tag{A.7}$$

is a curriculum region of γ if it holds: 1) $\Psi \wedge \mathbf{v} \in [0,1]^n$ is nonempty; 2) $a_i < a_j$ for all $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$; $a_i = a_j$ for all $\gamma(\mathbf{x}_i) = \gamma(\mathbf{x}_j)$.

Proof. (1) $\Psi \wedge \mathbf{v} \in [0,1]^n$ is a nonempty convex set.

(2) For $\mathbf{x}_i, \mathbf{x}_j$ with $\gamma(\mathbf{x}_i) < \gamma(\mathbf{x}_j)$, denote $\Psi_{ij} = \{\mathbf{v}_{ij} | \mathbf{a}_{ij}^T \mathbf{v}_{ij} \leq c\}$, $\mathbf{a}_{ij}/\mathbf{v}_{ij}$ the sub-vector of $\mathbf{a}/\mathbf{v}$ by wiping off its ith and jth elements, respectively, we can then calculate the expected value of v_i on the region $\Psi = \{\mathbf{v} | \mathbf{a}^T \mathbf{v} \leq c\}$ as:

$$\begin{aligned}
E(v_i) &= \int_{\Psi} v_i \, d\mathbf{v} \\
&= \int_{\Psi_{ij}} \int_0^{\frac{c-\mathbf{a}_{ij}^T \mathbf{v}_{ij}}{a_j}} \int_0^{\frac{c-\mathbf{a}_{ij}^T \mathbf{v}_{ij} - a_j v_j}{a_i}} v_i dv_i dv_j d\mathbf{v}_{ij} \\
&= \int_{\Psi_{ij}} \int_0^{\frac{c-\mathbf{a}_{ij}^T \mathbf{v}_{ij}}{a_j}} \frac{\left(c - \mathbf{a}_{ij}^T \mathbf{v}_{ij} - a_j v_j\right)^2}{2a_i^2} dv_j d\mathbf{v}_{ij} \\
&= \frac{\int_{\Psi_{ij}} \left(c - \mathbf{a}_{ij}^T \mathbf{v}_{ij}\right)^3 d\mathbf{v}_{ij}}{6a_i^2 a_j}.
\end{aligned}$$

In the similar way, we can get that:

$$E(v_j) = \int_{\Psi} v_j \, d\mathbf{v} = \frac{\int_{\Psi_{ij}} \left(c - \mathbf{a}_{ij}^T \mathbf{v}_{ij}\right)^3 d\mathbf{v}_{ij}}{6a_j^2 a_i}.$$

We thus can get that

$$E(v_i) - E(v_j) = \frac{\int_{\Psi_{ij}} \left(c - \mathbf{a}_{ij}^T \mathbf{v}_{ij}\right)^3 d\mathbf{v}_{ij}}{6a_i^2 a_j^2} (a_j - a_i) > 0.$$

Similarly, we can prove that $\int_{\Psi} v_i \, d\Psi = \int_{\Psi} v_j \, d\Psi$ for $\gamma(\mathbf{x}_i) = \gamma(\mathbf{x}_j)$.

The proof is then completed. □

Theorem 7.6: The binary, linear, logarithmic and mixture scheme are self-paced functions.

Proof. We first prove the above functions satisfying Condition 1 in Definition 7.4, i.e. they are convex with respect to $\mathbf{v} \in [0,1]^n$, where n is the number of samples. As binary, linear, logarithmic and mixture self-paced functions can be decoupled $f(\mathbf{v}; \lambda) = \sum_{i=1}^n f(v_i; \lambda)$:

For binary scheme $f(v_i;\lambda) = -\lambda v_i$:

$$\frac{\partial^2 f}{\partial^2 v_i} = 0. \tag{A.8}$$

For linear scheme $f(v_i;\lambda) = \frac{1}{2}\lambda(v_i^2 - 2v_i)$:

$$\frac{\partial^2 f}{\partial^2 v_i} = \lambda > 0, \tag{A.9}$$

where $\lambda > 0$.

For logarithmic scheme $f(v_i;\lambda) = \zeta v_i - \frac{\zeta^{v_i}}{\log \zeta}$:

$$\frac{\partial^2 f}{\partial^2 v_i} = -\frac{1}{\log \zeta}\zeta^{v_i} > 0, \tag{A.10}$$

where $\zeta = 1 - \lambda$ and $\lambda \in (0, 1)$.

For mixture scheme $f(v_i;\lambda) = -\zeta \log(v_i + \frac{1}{\lambda_1}\zeta)$:

$$\frac{\partial^2 f}{\partial^2 v_i} = \frac{\zeta \lambda_1^2}{(\zeta + \lambda_1 v_i)^2} > 0 \tag{A.11}$$

where $\lambda = [\lambda_1, \lambda_2]$, $\zeta = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2}$, and $\lambda_1 > \lambda_2 > 0$.

As the above second derivatives are non-negative, and the sum of convex functions is convex, we have $f(\mathbf{v};\lambda)$ for binary, linear, logarithmic and mixture scheme are convex.

We then prove the above functions satisfying Condition 2 that is when all variables are fixed except for v_i, ℓ_i, v_i^* decreases with ℓ_i

Denote $\mathbb{E}_{\mathbf{w}} = \sum_{i=1}^{n} v_i \ell_i + f(\mathbf{v};\lambda)$ as the objective with the fixed model parameters $\mathbf{w}$, where ℓ_i is the loss for the i^{th} sample. The optimal solution $\mathbf{v}^* = [v_1^*, \cdots, v_n^*]^T = \arg\min_{\mathbf{v}\in[0,1]^n} \mathbb{E}_{\mathbf{w}}$.

For binary scheme:

$$\begin{aligned} \mathbb{E}_{\mathbf{w}} &= \sum_{i=1}^{n} (\ell_i - \lambda) v_i; \\ \frac{\partial \mathbb{E}_{\mathbf{w}}}{\partial v_i} &= \ell_i - \lambda = 0; \\ \Rightarrow v_i^* &= \begin{cases} 1 & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda. \end{cases} \end{aligned} \tag{A.12}$$

For linear scheme:

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}} &= \sum_{i=1}^{n} \ell_i v_i + \frac{1}{2}\lambda(v_i^2 - 2v_i); \\
\frac{\partial \mathbb{E}_{\mathbf{w}}}{\partial v_i} &= \ell + v_i\lambda - \lambda = 0; \\
\Rightarrow v_i^* &= \begin{cases} -\frac{1}{\lambda}\ell + 1 & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda. \end{cases}
\end{aligned} \tag{A.13}$$

For logarithmic scheme:

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}} &= \sum_{i=1}^{n} \ell_i v_i + \zeta v_i - \frac{\zeta^{v_i}}{\log \zeta}; \\
\frac{\partial \mathbb{E}_{\mathbf{w}}}{\partial v_i} &= \ell + \zeta - \zeta^{v_i} = 0; \\
\Rightarrow v_i^* &= \begin{cases} \frac{1}{\log \zeta} \log(\ell + \zeta) & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda. \end{cases}
\end{aligned} \tag{A.14}$$

where $\zeta = 1 - \lambda$ $(0 < \lambda < 1)$.

For mixture scheme:

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}} &= \sum_{i=1}^{n} \ell_i v_i - \zeta \log(v_i + \frac{1}{\lambda_1}\zeta); \\
\frac{\partial \mathbb{E}_{\mathbf{w}}}{\partial v_i} &= \ell - \frac{\zeta\lambda_1}{\zeta + \lambda_1 v_i} = 0; \\
\Rightarrow v_i^* &= \begin{cases} 1 & \ell_i \leq \lambda_2 \\ 0 & \ell_i \geq \lambda_1 \\ \frac{(\lambda_1 - \ell)\zeta}{\ell\lambda_1} & \lambda_2 < \ell_i < \lambda_1 \end{cases}
\end{aligned} \tag{A.15}$$

where $\lambda = [\lambda_1, \lambda_2]$, and $\zeta = \frac{\lambda_1\lambda_2}{\lambda_1 - \lambda_2}$, $(\lambda_1 > \lambda_2 > 0)$.

By setting the partial gradient to zero we arrive the optimal solution of $\mathbf{v}$. It is obvious that v_i is decreasing with respect to ℓ_i in all functions. In all cases, we have that $\lim\limits_{\ell_i \to 0} v_i^* = 1, \lim\limits_{\ell_i \to \infty} v_i^* = 0$.

Finally, we prove that the above functions satisfying Condition 3 that is $\|\mathbf{v}\|_1$ increases with respect to λ, and it holds that $\forall i \in [1, n]$, $\lim\limits_{\lambda \to 0} v_i^* = 0, \lim\limits_{\lambda \to \infty} v_i^* = 1$.

It is easy to verify that each individual v_i^* increases with respect to λ in their closed-form solutions in Eq. (A.12), Eq. (A.13), Eq. (A.14) and Eq. (A.15) (in mixture scheme, let $\lambda = \lambda_1$ represent the model age). Therefore $\|\mathbf{v}\|_1 = \sum_{i=1}^{n} v_i$ also increases with respect to λ. In an extreme case, when λ approaches positive infinity, we have $\forall i \in [1, n] v_i = 1$, i.e. $\lim_{\lambda\to\infty} v_i^* = 1$ in Eq. (A.12), Eq. (A.13), Eq. (A.14) and Eq. (A.15). Similarly, when λ approaches 0, we have $\lim_{\lambda\to 0} v_i^* = 0$.

As binary, linear, logarithmic and mixture scheme satisfy the three conditions, they are all self-paced functions.

The proof is then completed.

□

***Theorem* 7.7**: *Algorithm 3 attains the global optimum to* $\min_{\mathbf{v}} \mathbb{E}(\mathbf{w}, \mathbf{v})$ *for any given* $\mathbf{w}$ *in linearithmic time.*

Proof. Given the training dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^m$ denotes the i^{th} observed sample and y_i denotes its label. Assume that the training samples $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$ are with b groups: $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(b)}$, where $\mathbf{X}^{(j)} = (\mathbf{x}_1^{(j)}, \cdots, \mathbf{x}_{n_j}^{(j)}) \in \mathbb{R}^{m\times n_j}$ corresponds to samples in the j^{th} group, n_j is the sample number in this group and $\sum_{j=1}^{b} n_j = n$. Accordingly, denote the weight vector as $\mathbf{v} = [\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(b)}]$, where $\mathbf{v}^{(j)} = (v_1^{(j)}, \cdots, v_{n_j}^{(j)})^T \in \mathbb{R}^{n_j}$. The following theorem proves that Algorithm 1 can get the global solution of the following non-convex optimization problem:

$$\min_{\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda, \gamma) = \sum_{i=1}^{n} v_i L(y_i, f(\mathbf{x}_i, \mathbf{w})) - \lambda \sum_{i=1}^{n} v_i - \gamma \|\mathbf{v}\|_{2,1}, \tag{A.16}$$

where $L(y_i, f(\mathbf{x}_i, \mathbf{w}))$ denotes the loss function which calculates the cost between the ground truth label y_i and the estimated label $f(\mathbf{x}_i, \mathbf{w})$, and the $l_{2,1}$-norm $\|\mathbf{v}\|_{2,1}$ is the group sparsity of $\mathbf{v}$:

$$\|\mathbf{v}\|_{2,1} = \sum_{j=1}^{b} \|\mathbf{v}^{(j)}\|_2.$$

For convenience we briefly rewrite $\mathbb{E}(\mathbf{w}, \mathbf{v}; \lambda, \gamma)$ and $L(y_i, f(\mathbf{x}_i, \mathbf{w}))$ as $\mathbb{E}(\mathbf{v})$ and L_i, respectively.

The weight vector $\mathbf{v}^*$ outputted from Algorithm 1 attains the global optimal solution of the optimization problem (A.16), i.e.,

$$\mathbf{v}^* = \arg \min_{\mathbf{v}\in[0,1]^n} \mathbb{E}(\mathbf{v}).$$

The objective function of (A.16) can be reformulated as the following decoupling forms based on the data cluster information:

$$\mathbb{E}(\mathbf{v}) = \sum_{j=1}^{b} E(\mathbf{v}^{(j)}), \tag{A.17}$$

where

$$E(\mathbf{v}^{(j)}) = \sum_{i=1}^{n_j} v_i^{(j)} L_i^{(j)} - \lambda \sum_{i=1}^{n_j} v_i^{(j)} - \gamma \|\mathbf{v}^{(j)}\|_2, \tag{A.18}$$

where $L_i^{(j)}$ represents the loss value of $\mathbf{x}_i^{(j)}$. It is easy to see that the original problem (A.16) can be equivalently decomposed as a series of the following sub-optimization problems ($j = 1, \cdots, b$):

$$\mathbf{v}^{(j)^*} = \arg \min_{\mathbf{v}^{(j)} \in [0,1]^{n_j}} \mathbb{E}(\mathbf{v}^{(j)}). \tag{A.19}$$

$E(\mathbf{v}^{(j)})$ defined in Eq. (A.18) is a concave function since its first and second terms are linear, and the third term is the negative $l_{2,1}$ norm, whose positive form is a commonly utilized convex regularizer. It is well known that the minimum solution of a concave function over a polytope can be obtained at its vertices [187]. In other words, for the optimization problem (A.19), it holds that its optimal solution $\mathbf{v}^{(j)^*} \in \{0,1\}^{n_j}$, i.e.,

$$\mathbf{v}^{(j)^*} = \arg \min_{\mathbf{v}^{(j)} \in \{0,1\}^{n_j}} \mathbb{E}(\mathbf{v}^{(j)}). \tag{A.20}$$

For $k = 1, \cdots, n_j$, let's denote

$$\mathbf{v}^{(j)}(k) = \arg \min_{\substack{\mathbf{v}^{(j)} \in \{0,1\}^{n_j} \\ \|\mathbf{v}^{(j)}\|_0 = k}} \mathbb{E}(\mathbf{v}^{(j)}). \tag{A.21}$$

This means that $\mathbf{v}^{(j)}(k)$ is the optimum of (A.19) if it is further constrained to be with k nonzero entries. It is then easy to deduce that

$$\mathbf{v}^{(j)^*} = \arg \min_{\mathbf{v}^{(j)}(k)} \mathbb{E}(\mathbf{v}^{(j)}(k)). \tag{A.22}$$

That is, the optimal solution $\mathbf{v}^{(j)^*}$ of (A.19) can be achieved among $\mathbf{v}^{(j)}(1), \cdots, \mathbf{v}^{(j)}(n_j)$ at which the minimal objective value is attained.

Without loss of generality, we assume that the samples $(\mathbf{x}_1^{(j)}, \cdots, \mathbf{x}_{n_j}^{(j)})$ in the j^{th} cluster are arranged in the ascending order of their loss values $L_i^{(j)}$. Then for the optimization problem (A.21), we can get that

$$\min_{\substack{\mathbf{v}^{(j)} \in \{0,1\}^{n_j} \\ \|\mathbf{v}^{(j)}\|_0 = k}} \mathbb{E}(\mathbf{v}^{(j)}) = \sum_{i=1}^{n_j} v_i^{(j)} L_i^{(j)} - \lambda \sum_{i=1}^{n_j} v_i^{(j)} - \gamma \|\mathbf{v}^{(j)}\|_2$$

$$\Leftrightarrow \quad \min_{\substack{\mathbf{v}^{(j)} \in \{0,1\}^{n_j} \\ \|\mathbf{v}^{(j)}\|_0 = k}} \sum_{i=1}^{n_j} v_i^{(j)} L_i^{(j)},$$

since the last two terms in $\mathbb{E}(\mathbf{v}^{(j)})$ are with constant values under the constraint. Then it is easy to get that the optimal solution $\mathbf{v}^{(j)}(k)$ of (A.21) is attained by setting its k entries corresponding to the k smallest loss values $L_i^{(j)}$ (i.e., the first k entries of $\mathbf{v}^{(j)}(k)$) as 1 while others as 0, and the minimal objective value is

$$E(\mathbf{v}^{(j)}(k)) = \sum_{i=1}^{k} v_i^{(j)} L_i^{(j)} - \lambda k - \gamma\sqrt{k}. \tag{A.23}$$

Then let's calculate the difference between any two adjacent elements in the sequence $E(\mathbf{v}^{(j)}(1)), \cdots, E(\mathbf{v}^{(j)}(n_j))$:

$$\begin{aligned} diff_k &= E(\mathbf{v}^{(j)}(k+1)) - E(\mathbf{v}^{(j)}(k)) \\ &= L_{k+1}^{(j)} - \lambda - \gamma(\sqrt{k+1} - \sqrt{k}) \\ &= L_{k+1}^{(j)} - (\lambda + \gamma\frac{1}{\sqrt{k+1}+\sqrt{k}}). \end{aligned}$$

Since $L_k^{(j)}$ (with respect to k) is a monotonically increasing sequence while $\lambda+\gamma\frac{1}{\sqrt{k+1}+\sqrt{k}}$ is a monotonically decreasing sequence, $diff_k$ is a monotonically increasing sequence. Denote k^* as the index where its first positive value is attained (if $diff_k \leq 0$ for all $k = 1, \cdots, n_j - 1$, $k^* = n_j$). Then it is easy to get that $E(\mathbf{v}^{(j)}(k))$ is monotonically decreasing until $k = k^*$ and then it starts to be monotonically increasing. This means that $E(\mathbf{v}^{(j)}(k^*))$ gets the minimum among all $E(\mathbf{v}^{(j)}(1)), \cdots, E(\mathbf{v}^{(j)}(n_j))$. Based on (A.22), we know that the global optimum $\mathbf{v}^{(j)^*}$ of (A.19) is attained at $\mathbf{v}^{(j)}(k^*)$.

By independently calculating the optimum $\mathbf{v}^{(j)^*}$ for each cluster and then combining them, the global optimal solution $\mathbf{v}^*$ of (A.16) can then be calculated. This corresponds to the process of our proposed Algorithm 3.

The most computational complex step in the above derivation is the sort of n_j $(1 \leq j \leq b)$ samples. Since $n_j < n$, the average-case complexity is thus upper bounded by $O(n \log n)$, assuming that the quick sort algorithm is used.

The proof is completed. □

Appendix B

Detailed Results

Table B.1: Event-level comparison of AP on the 10 splits of MED13Test.

Event ID & Name	Raw + Expert	Adjusted + Expert	Adjusted + Auto	Adjusted + AutoVisual
E006: Birthday party	0.3411	0.2980	0.1207	0.1207
E007: Changing a vehicle tire	0.0967	0.1667	0.2061	0.0134
E008: Flash mob gathering	0.2087	0.1647	0.1028	0.1028
E009: Getting a vehicle unstuck	0.1416	0.1393	0.0569	0.0569
E010: Grooming an animal	0.0442	0.0479	0.0128	0.0128
E011: Making a sandwich	0.0909	0.0804	0.2910	0.0709
E012: Parade	0.4552	0.4685	0.2027	0.2027
E013: Parkour	0.0498	0.0596	0.0619	0.0525
E014: Repairing an appliance	0.2731	0.2376	0.2262	0.0234
E015: Working on a sewing project	0.2022	0.2184	0.0135	0.0045
E021: Attempting a bike trick	0.0969	0.1163	0.0486	0.0486
E022: Cleaning an appliance	0.1248	0.1248	0.1248	0.0124
E023: Dog show	0.7284	0.7288	0.6028	0.6027
E024: Giving directions to a location	0.0253	0.0252	0.0252	0.0069
E025: Marriage proposal	0.0748	0.0750	0.0755	0.0011
E026: Renovating a home	0.0139	0.0061	0.0049	0.0049
E027: Rock climbing	0.1845	0.1724	0.0668	0.0668
E028: Town hall meeting	0.1585	0.0898	0.0163	0.0163
E029: Winning a race without a vehicle	0.1470	0.1697	0.0584	0.0584
E030: Working on a metal crafts project	0.0673	0.0422	0.0881	0.0026
MAP	0.1762	0.1716	0.1203	0.0741

Table B.2: Event-level comparison of AP on the 10 splits of MED14Test.

Event ID & Name	Raw + Expert	Adjusted + Expert	Adjusted + Auto	Adjusted + AutoVisual
E021: Attempting a bike trick	0.0632	0.0678	0.0814	0.0822
E022: Cleaning an appliance	0.2634	0.2635	0.2634	0.2636
E023: Dog show	0.6757	0.6449	0.4387	0.4414
E024: Giving directions to a location	0.0613	0.0613	0.0614	0.0612
E025: Marriage proposal	0.0176	0.0174	0.0181	0.0174
E026: Renovating a home	0.0252	0.0089	0.0043	0.0043
E027: Rock climbing	0.2082	0.1302	0.0560	0.0560
E028: Town hall meeting	0.2478	0.0925	0.0161	0.0161
E029: Winning a race without a vehicle	0.1234	0.1848	0.0493	0.0497
E030: Working on a metal crafts project	0.1238	0.0616	0.0981	0.0981
E031: Beekeeping	0.5900	0.5221	0.4217	0.4258
E032: Wedding shower	0.0834	0.0924	0.0922	0.0395
E033: Non-motorized vehicle repair	0.5218	0.4525	0.0149	0.0150
E034: Fixing musical instrument	0.0284	0.0439	0.0439	0.0023
E035: Horse riding competition	0.3673	0.3346	0.0994	0.0993
E036: Felling a tree	0.0970	0.0620	0.0108	0.0108
E037: Parking a vehicle	0.2921	0.2046	0.0313	0.0313
E038: Playing fetch	0.0339	0.0284	0.0016	0.0014
E039: Tailgating	0.1429	0.0200	0.0010	0.0010
E040: Tuning musical instrument	0.1553	0.1553	0.1840	0.0128
MAP	0.2061	0.1724	0.0994	0.0865

Table B.3: Performance for 30 commercials on the YFCC100 set.

ID	Query Name	Commercial Product	Evaluation Metric			Category
			P@20	MRR	MAP@20	
1	football_and_running	soccer shoes	0.80	1.00	0.88	Sport
2	auto_racing	sport cars	0.70	1.00	0.91	Auto
3	dog_show	dog training collars	0.95	1.00	0.97	Grocery
4	baby	stroller/diapper	1.00	1.00	1.00	Grocery
5	fire_burning_smoke	fire prevention	0.95	1.00	0.96	Miscellaneous
6	cake_or_birthday_cake	birthday cake	0.35	0.50	0.60	Grocery
7	underwater	diving	1.00	1.00	1.00	Sports
8	dog_indoor	dog food	0.75	1.00	0.67	Grocery
9	riding_horse	horse riding lessons	0.90	1.00	0.93	Sports
10	kitchen_food	restaurant	1.00	1.00	1.00	Grocery
11	Christmas_decoration	decoration	0.80	1.00	0.87	Grocery
12	dancing	dancing lessons	0.90	1.00	0.90	Miscellaneous
13	bicycling	cycling cloth and helmet	0.95	1.00	0.99	Sports
14	car_and_vehicle	car tires	1.00	1.00	1.00	Auto
15	skiing_or_snowboarding	ski resort	0.95	1.00	0.96	Sports
16	parade	flags or banners	0.90	1.00	0.96	Grocery
17	music_band	live music show	1.00	1.00	1.00	Grocery
18	busking	live show	0.20	1.00	0.50	Miscellaneous
19	home_renovation	furniture	0.00	0.00	0.00	Miscellaneous
20	speaking_in_front_of_people	speaking in public training	0.65	0.50	0.63	Miscellaneous
21	sunny_beach	vacation by beach	1.00	1.00	1.00	Traveling
22	politicians	vote Obama	0.60	1.00	0.63	Miscellaneous
23	female_face	makeup	1.00	1.00	1.00	Miscellaneous
24	cell_phone	cell phone	0.80	1.00	0.96	Miscellaneous
25	fireworks	fireworks	0.95	1.00	0.96	Miscellaneous
26	tennis	tennis	1.00	1.00	1.00	Sports
27	helicopter	helicopter tour	1.00	1.00	1.00	Traveling
28	cooking	pan	0.90	1.00	0.92	Miscellaneous
29	eiffel_night	hotels in Paris	0.90	1.00	0.89	Traveling
30	table_tennis	ping pong	0.60	1.00	0.85	Sports

Table B.4: Event-level comparison of modality contribution on the NIST split. The best AP is marked in bold.

Event ID & Name	FullSys	FullSys+PRF	VisualSys	ASRSys	OCRSys
E006: Birthday party	0.3842	**0.3862**	0.3673	0.0327	0.0386
E007: Changing a vehicle tire	0.2322	**0.3240**	0.2162	0.1707	0.0212
E008: Flash mob gathering	0.2864	**0.4310**	0.2864	0.0052	0.0409
E009: Getting a vehicle unstuck	**0.1588**	0.1561	**0.1588**	0.0063	0.0162
E010: Grooming an animal	**0.0782**	0.0725	**0.0782**	0.0166	0.0050
E011: Making a sandwich	0.1183	0.1304	0.1064	**0.2184**	0.0682
E012: Parade	**0.5566**	0.5319	**0.5566**	0.0080	0.0645
E013: Parkour	0.0545	**0.0839**	0.0448	0.0043	0.0066
E014: Repairing an appliance	0.2619	**0.2989**	0.2341	0.2086	0.0258
E015: Working on a sewing project	**0.2068**	0.2021	**0.2036**	0.0866	0.0166
E021: Attempting a bike trick	0.0635	**0.0701**	0.0635	0.0006	0.0046
E022: Cleaning an appliance	**0.2634**	0.1747	0.0008	**0.2634**	0.0105
E023: Dog show	**0.6737**	0.6610	**0.6737**	0.0009	0.0303
E024: Giving directions to a location	**0.0614**	0.0228	0.0011	**0.0614**	0.0036
E025: Marriage proposal	0.0188	**0.0270**	0.0024	0.0021	0.0188
E026: Renovating a home	**0.0252**	0.0160	**0.0252**	0.0026	0.0023
E027: Rock climbing	**0.2077**	0.2001	0.2077	0.1127	0.0038
E028: Town hall meeting	0.2492	**0.3172**	0.2492	0.0064	0.0134
E029: Winning a race without a vehicle	0.1257	**0.1929**	0.1257	0.0011	0.0019
E030: Working on a metal crafts project	0.1238	**0.1255**	0.0608	0.0981	0.0142
E031: Beekeeping	0.5883	**0.6401**	0.5883	0.2676	0.0440
E032: Wedding shower	0.0833	**0.0879**	0.0459	0.0428	0.0017
E033: Non-motorized vehicle repair	0.5198	**0.5263**	0.5198	0.0828	0.0159
E034: Fixing musical instrument	0.0276	**0.0444**	0.0170	0.0248	0.0023
E035: Horse riding competition	0.3677	**0.3710**	0.3677	0.0013	0.0104
E036: Felling a tree	0.0968	**0.1180**	0.0968	0.0020	0.0076
E037: Parking a vehicle	**0.2918**	0.2477	**0.2918**	0.0008	0.0009
E038: Playing fetch	0.0339	**0.0373**	0.0339	0.0020	0.0014
E039: Tailgating	0.1437	**0.1501**	0.1437	0.0013	0.0388
E040: Tuning musical instrument	0.1554	**0.3804**	0.0010	0.1840	0.0677
MAP (MED13Test E006-E015 E021-E030)	0.2075	**0.2212**	0.1831	0.0653	0.0203
MAP (MED14Test E021-E040)	0.2060	**0.2205**	0.1758	0.0579	0.0147

Table B.5: Event-level comparison of visual feature contribution on the NIST split.

Event ID & Name	FullSys	MED/IACC	MED/Sports	MED/YFCC	MED/DIY	MED/ImageNet
E006: Birthday party	0.3842	0.3797	0.3842	0.2814	0.3842	0.2876
E007: Changing a vehicle tire	0.2322	0.2720	0.2782	0.1811	0.1247	0.0998
E008: Flash mob gathering	0.2864	0.1872	0.2864	0.3345	0.2864	0.2864
E009: Getting a vehicle unstuck	0.1588	0.1070	0.1588	0.1132	0.1588	0.1588
E010: Grooming an animal	0.0782	0.0902	0.0782	0.0914	0.0474	0.0782
E011: Making a sandwich	0.1183	0.0926	0.1183	0.1146	0.1183	0.1183
E012: Parade	0.5566	0.5738	0.5566	0.3007	0.5566	0.5566
E013: Parkour	0.0545	0.0066	0.0545	0.0545	0.0545	0.0545
E014: Repairing an appliance	0.2619	0.2247	0.2619	0.1709	0.2619	0.1129
E015: Working on a sewing project	0.2068	0.2166	0.2068	0.2068	0.1847	0.0712
E021: Attempting a bike trick	0.0635	0.0635	0.0006	0.0635	0.0635	0.0635
E022: Cleaning an appliance	0.2634	0.2634	0.2634	0.2634	0.2634	0.2634
E023: Dog show	0.6737	0.6737	0.0007	0.6737	0.6737	0.6737
E024: Giving directions to a location	0.0614	0.0614	0.0614	0.0614	0.0614	0.0614
E025: Marriage proposal	0.0188	0.0188	0.0188	0.0188	0.0188	0.0188
E026: Renovating a home	0.0252	0.0017	0.0252	0.0252	0.0252	0.0252
E027: Rock climbing	0.2077	0.2077	0.0009	0.2077	0.2077	0.2077
E028: Town hall meeting	0.2492	0.0956	0.2492	0.2418	0.2492	0.2492
E029: Winning a race without a vehicle	0.1257	0.1257	0.0056	0.1257	0.1257	0.1257
E030: Working on a metal crafts project	0.1238	0.1238	0.1238	0.0981	0.1238	0.1238
E031: Beekeeping	0.5883	0.5883	0.5883	0.5883	0.5883	0.0012
E032: Wedding shower	0.0833	0.0833	0.0833	0.0833	0.0924	0.0833
E033: Non-motorized vehicle repair	0.5198	0.5198	0.4440	0.5198	0.4742	0.4417
E034: Fixing musical instrument	0.0276	0.0276	0.0276	0.0276	0.0439	0.0276
E035: Horse riding competition	0.3677	0.3430	0.1916	0.3677	0.3677	0.3677
E036: Felling a tree	0.0968	0.0275	0.1100	0.0968	0.0968	0.0968
E037: Parking a vehicle	0.2918	0.1902	0.2918	0.2918	0.2918	0.1097
E038: Playing fetch	0.0339	0.0339	0.0008	0.0339	0.0339	0.0339
E039: Tailgating	0.1437	0.0631	0.1437	0.0666	0.1437	0.1437
E040: Tuning musical instrument	0.1554	0.1554	0.1554	0.1554	0.1554	0.1554
MAP (MED13Test E006-E015 E021-E030)	0.2075	0.1893	0.1567	0.1814	0.1995	0.1818
MAP (MED14Test E021-E040)	0.2060	0.1834	0.1393	0.2005	0.2050	0.1637

TABLE B.6: Event-level comparison of textual feature contribution on the NIST split.

Event ID & Name	FullSys	MED/ASR	MED/OCR
E006: Birthday party	0.3842	0.3842	0.3673
E007: Changing a vehicle tire	0.2322	0.2162	0.2322
E008: Flash mob gathering	0.2864	0.2864	0.2864
E009: Getting a vehicle unstuck	0.1588	0.1588	0.1588
E010: Grooming an animal	0.0782	0.0782	0.0782
E011: Making a sandwich	0.1183	0.1043	0.1205
E012: Parade	0.5566	0.5566	0.5566
E013: Parkour	0.0545	0.0545	0.0448
E014: Repairing an appliance	0.2619	0.2436	0.2527
E015: Working on a sewing project	0.2068	0.1872	0.2242
E021: Attempting a bike trick	0.0635	0.0635	0.0635
E022: Cleaning an appliance	0.2634	0.0008	0.2634
E023: Dog show	0.6737	0.6737	0.6737
E024: Giving directions to a location	0.0614	0.0011	0.0614
E025: Marriage proposal	0.0188	0.0188	0.0024
E026: Renovating a home	0.0252	0.0252	0.0252
E027: Rock climbing	0.2077	0.2077	0.2077
E028: Town hall meeting	0.2492	0.2492	0.2492
E029: Winning a race without a vehicle	0.1257	0.1257	0.1257
E030: Working on a metal crafts project	0.1238	0.0608	0.1238
E031: Beekeeping	0.5883	0.5883	0.5883
E032: Wedding shower	0.0833	0.0833	0.0459
E033: Non-motorized vehicle repair	0.5198	0.5198	0.5198
E034: Fixing musical instrument	0.0276	0.0314	0.0178
E035: Horse riding competition	0.3677	0.3677	0.3677
E036: Felling a tree	0.0968	0.0968	0.0968
E037: Parking a vehicle	0.2918	0.2918	0.2918
E038: Playing fetch	0.0339	0.0339	0.0339
E039: Tailgating	0.1437	0.1437	0.1437
E040: Tuning musical instrument	0.1554	0.0893	0.1840
MAP (MED13Test E006-E015 E021-E030)	0.2075	0.1848	0.2059
MAP (MED14Test E021-E040)	0.2060	0.1836	0.2043

Bibliography

[1] John R Smith. Riding the multimedia big data wave. In *SIGIR*, 2013.

[2] James Davidson, Benjamin Liebald, Junning Liu, et al. The youtube video recommendation system. In *RecSys*, 2010.

[3] Baptist Vandersmissen, Fréderic Godin, Abhineshwar Tomar, Wesley De Neve, and Rik Van de Walle. The rise of mobile and social short-form video: an in-depth measurement study of vine. In *ICMR Workshop on Social Multimedia and Storytelling*, 2014.

[4] Lu Jiang, L Cao, Y Kalantidis, Sachin Farfade, Jiliang Tang, and Alexander G Hauptmann. Delving deep into personal photo and video search. In *WSDM*, 2017.

[5] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Quenot. TRECVID 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *NIST TRECVID*, 2014.

[6] Shicheng Xu, Huan Li, Xiaojun Chang, Shoou-I Yu, Xingzhong Du, Xuanchong Li, Lu Jiang, Zexi Mao, Zhenzhong Lan, Susanne Burger, et al. Incremental multimodal query construction for video search. In *ICMR*, 2015.

[7] Fiona Fui-Hoon Nah. A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology*, 23(3):153–163, 2004.

[8] Meng Wang, Richang Hong, Guangda Li, Zheng-Jun Zha, Shuicheng Yan, and Tat-Seng Chua. Event driven web video summarization by tag localization and key-shot identification. *Multimedia, IEEE Transactions on*, 14(4):975–985, 2012.

[9] Evlampios Apostolidis, Vasileios Mezaris, Mathilde Sahuguet, Benoit Huet, Barbora Červenková, Daniel Stein, Stefan Eickeler, José Luis Redondo Garcia, Raphaël Troncy, and Lukás Pikora. Automatic fine-grained hyperlinking of videos within a closed collection using scene segmentation. In *MM*, 2014.

[10] Yue Gao, Sicheng Zhao, Yang Yang, and Tat-Seng Chua. Multimedia social event detection in microblog. In *MMM*, 2015.

[11] Lu Jiang, Shoou-I Yu, Deyu Meng, Yi Yang, Teruko Mitamura, and Alexander G Hauptmann. Fast and accurate content-based semantic search in 100m internet videos. In *MM*, 2015.

[12] Fabricio Benevenuto, Tiago Rodrigues, Virgílio AF Almeida, Jussara Almeida, Marcos Gonçalves, and Keith Ross. Video pollution on the web. *First Monday*, 15(4), 2010.

[13] Zhuo Chen, Lu Jiang, Wenlu Hu, Kiryong Ha, Brandon Amos, Padmanabhan Pillai, Alex Hauptmann, and Mahadev Satyanarayanan. Early implementation experience with wearable cognitive assistance applications. In *Workshop on Wearable Systems and Applications in WearSys*, 2015.

[14] Andrei Broder, Lada Adamic, Michael Franklin, Maarten de Rijke, Eric Xing, and Kai Yu. Big data: New paradigm or sound and fury, signifying nothing? In *WSDM*, 2015.

[15] Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *MM*, 2014.

[16] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[18] Shoou-I Yu, Lu Jiang, Zhongwen Xu, et al. Informedia@ trecvid 2014 med and mer. In *TRECVID*, 2014.

[19] Yajie Miao, Lu Jiang, Hao Zhang, and Florian Metze. Improvements to speaker adaptive training of deep neural networks. In *SLT*, 2014.

[20] Yajie Miao, Mohammad Gowayyed, and Florian Metze. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. *arXiv preprint arXiv:1507.08240*, 2015.

[21] L. Jiang, A.G. Hauptmann, and G. Xiang. Leveraging high-level and low-level features for multimedia event detection. In *MM*, 2012.

[22] Yajie Miao. *Incorporating Context Information into Deep Neural Network Acoustic Models*. PhD thesis, Carnegie Mellon University, 2016.

[23] Jun Liu, Zhaohui Wu, Lu Jiang, Qinghua Zheng, and Xiao Liu. Crawling deep web content through query forms. *WEBIST*, 2009.

[24] Lu Jiang, Zhaohui Wu, Qian Feng, Jun Liu, and Qinghua Zheng. Efficient deep web crawling using reinforcement learning. In *PAKDD*, 2010.

[25] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, DTIC Document, 1996.

[26] Cai-Zhi Zhu and Shin'ichi Satoh. Large vocabulary quantization for searching instances from videos. In *ICMR*, 2012.

[27] Hervé Jégou, Florent Perronnin, Matthijs Douze, Javier Sanchez, Pablo Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *PAMI*, 34(9):1704–1716, 2012.

[28] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[29] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *MIR*, 2007.

[30] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[31] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, 2006.

[32] Josef Sivic and Andrew Zisserman. Video google: Efficient visual search of videos. In *Toward Category-Level Object Recognition*, 2006.

[33] Ondrej Chum, James Philbin, Andrew Zisserman, et al. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, volume 810, 2008.

[34] Xiao Wu, Alexander G Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. In *MM*, 2007.

[35] Milind Naphade, John R Smith, Jelena Tesic, Shih-Fu Chang, Winston Hsu, Lyndon Kennedy, Alexander Hauptmann, and Jon Curtis. Large-scale concept ontology for multimedia. *MultiMedia, IEEE*, 13(3):86–91, 2006.

[36] Milind R Naphade and John R Smith. On the detection of semantic concepts at trecvid. In *MM*, 2004.

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[38] Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *CVPR*, 2007.

[39] Yale Song, Louis-Philippe Morency, and Ronald W Davis. Action recognition by hierarchical sequence summarization. In *CVPR*, 2013.

[40] Manan Jain, Hervé Jégou, and Patrick Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013.

[41] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[42] Michael Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *CVPR*, 2009.

[43] Lu Jiang, Wei Tong, Deyu Meng, and Alexander G. Hauptmann. Towards efficient learning of optimal spatial bag-of-words representations. In *ICMR*, 2014.

[44] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[45] Alexander Hauptmann, Rong Yan, Wei-Hao Lin, Michael Christel, and Howard Wactlar. Can high-level concepts fill the semantic gap in video retrieval? a case study with broadcast news. *TMM*, 9(5):958–966, 2007.

[46] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[48] Balakrishnan Varadarajan, George Toderici, Sudheendra Vijayanarasimhan, and Apostol Natsev. Efficient large scale video classification. *arXiv preprint arXiv:1505.06250*, 2015.

[49] Junwei Liang, Lu Jiang, Deyu Meng, and Alexander Hauptmann. Learning to detect concepts from webly-labeled video data. In *IJCAI*, 2016.

[50] Paul Over, George M Awad, Jon Fiscus, Brian Antonishek, Martial Michel, Alan F Smeaton, Wessel Kraaij, and Georges Quénot. Trecvid 2010–an overview of the goals, tasks, data, evaluation mechanisms, and metrics. In *NIST TRECVID*, 2010.

[51] Shoou-I Yu, Lu Jiang, Zhongwen Xu, Yi Yang, and Alexander G Hauptmann. Content-based video search over 1 million videos with 1 core in 1 second. In *ICMR*, 2015.

[52] Sangmin Oh, Scott McCloskey, Ilseo Kim, Arash Vahdat, Kevin J Cannons, Hossein Hajimirsadeghi, Greg Mori, AG Amitha Perera, Megha Pandey, and Jason J Corso. Multimedia event detection with multimodal feature fusion and temporal concept localization. *Machine vision and applications*, 25(1):49–69, 2014.

[53] Amirhossein Habibian, Thomas Mensink, and Cees GM Snoek. Composite concept discovery for zero-shot video event detection. In *ICMR*, 2014.

[54] Masoud Mazloom, Xirong Li, and Cees GM Snoek. Few-example video event retrieval using tag propagation. In *ICMR*, 2014.

[55] Shuang Wu, Sravanthi Bondugula, Florian Luisier, Xiaodan Zhuang, and Pradeep Natarajan. Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *CVPR*, 2014.

[56] Lu Jiang, Teruko Mitamura, Shoou-I Yu, and Alexander G Hauptmann. Zero-example event search using multimodal pseudo relevance feedback. In *ICMR*, 2014.

[57] Hyungtae Lee. Analyzing complex events and human actions in" in-the-wild" videos. In *UMD Ph.D Theses and Dissertations*, 2014.

[58] Lu Jiang, Shoou-I Yu, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *ICMR*, 2015.

[59] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[60] Nikolaos Gkalelis and Vasileios Mezaris. Video event detection using generalized subclass discriminant analysis and linear support vector machines. In *ICMR*, 2014.

[61] Feng Wang, Zhanhu Sun, Y Jiang, and C Ngo. Video event detection using motion relativity and feature selection. In *TMM*, 2013.

[62] Amirhossein Habibian, Koen EA van de Sande, and Cees GM Snoek. Recommendations for video event recognition using concept vocabularies. In *ICMR*, 2013.

[63] Bahjat Safadi, Mathilde Sahuguet, and Benoit Huet. When textual and visual information join forces for multimedia retrieval. In *ICMR*, 2014.

[64] Subhabrata Bhattacharya, Felix X Yu, and Shih-Fu Chang. Minimally needed evidence for complex event recognition in unconstrained videos. In *ICMR*, 2014.

[65] Wei Tong, Yi Yang, Lu Jiang, et al. E-LAMP: integration of innovative ideas for multimedia event detection. *Machine Vision and Applications*, 25(1):5–15, 2014.

[66] Amirhossein Habibian, Thomas Mensink, and Cees GM Snoek. Videostory: A new multimedia embedding for few-example recognition and translation of events. In *MM*, 2014.

[67] Ethem F Can and R Manmatha. Modeling concept dependencies for event detection. In *ICMR*, 2014.

[68] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *ECCV*, 2014.

[69] Michael Grant, Stephen Boyd, and Yinyu Ye. CVX: Matlab software for disciplined convex programming, 2008.

[70] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[71] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[72] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[73] Marshall L Fisher. The lagrangian relaxation method for solving integer programming problems. *Management science*, 50(12):1861–1871, 2004.

[74] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[75] Xiaofan Xu, Malay Ghosh, et al. Bayesian variable selection and estimation for group lasso. *Bayesian Analysis*, 10(4):909–936, 2015.

[76] Stephen E Robertson, Steve Walker, Micheline Beaulieu, and Peter Willett. Okapi at trec-7: automatic ad hoc, filtering, vlc and interactive track. In *NIST TREC*, 1999.

[77] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander G Hauptmann. Self-paced learning with diversity. In *NIPS*, 2014.

[78] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *AAAI*, 2015.

[79] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[80] Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. Instructional videos for unsupervised harvesting and learning of action examples. In *MM*, 2014.

[81] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, et al. The kaldi speech recognition toolkit. In *ASRU*, 2011.

[82] Erik Hatcher and Otis Gospodnetic. Lucene in action. In *Manning Publications*, 2004.

[83] Jeffrey Dalton, James Allan, and Pranav Mirajkar. Zero-shot video retrieval using content and concepts. In *CIKM*, 2013.

[84] Bharat Singh, Xintong Han, Zhe Wu, Vlad I Morariu, and Larry S Davis. Selecting relevant web trained concepts for automated event retrieval. *arXiv preprint arXiv:1509.07845*, 2015.

[85] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *ACL*, 1994.

[86] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *CICLing*, 2002.

[87] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[88] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[89] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *ACL*, 2014.

[90] Ehsan Younessian, Teruko Mitamura, and Alexander Hauptmann. Multimodal knowledge-based analysis in multimedia event detection. In *ICMR*, 2012.

[91] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[92] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *TOIS*, 22(2), 2004.

[93] Lu Jiang, Yajie Miao, Yi Yang, Zhenzhong Lan, and Alexander G Hauptmann. Viral video style: A closer look at viral videos on youtube. In *ICMR*, 2014.

[94] Yajie Miao, Florian Metze, and Seema Rawat. Deep maxout networks for low-resource speech recognition. In *ASRU*, 2013.

[95] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[96] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, Ricardo Baeza-Yates, Andrew Feng, Erik Ordentlich, Lee Yang, and Gavin Owens. Scalable semantic matching of queries to ads in sponsored search advertising. In *SIGIR*, 2016.

[97] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2013.

[98] Yi Wu, Edward Y Chang, Kevin Chen-Chuan Chang, and John R Smith. Optimal multimodal fusion for multimedia data analysis. In *MM*, 2004.

[99] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *MM*, 2010.

[100] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[101] Adish Singla and Ryen W White. Sampling high-quality clicks from noisy click data. In *WWW*, 2010.

[102] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[103] Pierre Garrigues, Sachin Farfade, Hamid Izadinia, Kofi Boakye, and Yannis Kalantidis. Tag prediction at flickr: a view from the darkroom. *LSCVS NIPS Workshop*, 2016.

[104] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[105] Martın Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[106] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2011.

[107] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, 2001.

[108] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[109] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4): 321–377, 1936.

[110] Shih-Fu Chang. How far we've come: Impact of 20 years of multimedia information retrieval. *TOMCCAP*, 9(1):42, 2013.

[111] Josip Krapac, Moray Allan, Jakob Verbeek, and Frédéric Jurie. Improving web image search results using query-relative classifiers. In *CVPR*, 2010.

[112] Xinmei Tian, Yijuan Lu, Linjun Yang, and Qi Tian. Learning to judge image search results. In *MM*, 2011.

[113] Nobuyuki Morioka and Jingdong Wang. Robust visual reranking via sparsity and ranking constraints. In *MM*, 2011.

[114] Victor Lavrenko and W Bruce Croft. Relevance based language models. In *SIGIR*, 2001.

[115] Winston H Hsu, Lyndon S Kennedy, and Shih-Fu Chang. Video search reranking via information bottleneck principle. In *MM*, 2006.

[116] Rong Yan, Alexander G Hauptmann, and Rong Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *MM*, 2003.

[117] Alexander G Hauptmann, Michael G Christel, and Rong Yan. Video retrieval based on semantic concepts. *Proceedings of the IEEE*, 96(4):602–622, 2008.

[118] Rong Yan, Alexander G Hauptmann, and Rong Jin. Multimedia search with pseudo-relevance feedback. In *CVIR*, pages 238–247, 2003.

[119] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

[120] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NIPS*, 2010.

[121] Yuan Liu, Tao Mei, Xian-Sheng Hua, Jinhui Tang, Xiuqing Wu, and Shipeng Li. Learning to video search rerank via pseudo preference feedback. In *ICME*, 2008.

[122] Jiawei Chen, Yin Cui, Guangnan Ye, Dong Liu, and Shih-Fu Chang. Event-driven semantic concept discovery by exploiting weakly tagged internet images. In *ICMR*, 2014.

[123] Jeffrey Dalton, James Allan, and Pranav Mirajkar. Zero-shot video retrieval using content and concepts. In *CIKM*, 2013.

[124] Kyung Soon Lee, W Bruce Croft, and James Allan. A cluster-based resampling method for pseudo-relevance feedback. In *SIGIR*, 2008.

[125] Guihong Cao, Jian Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR*, 2008.

[126] Xinmei Tian, Linjun Yang, Jingdong Wang, Yichen Yang, Xiuqing Wu, and Xian-Sheng Hua. Bayesian video search reranking. In *MM*, 2008.

[127] Ionut Mironica, Bogdan Ionescu, Jasper Uijlings, and Nicu Sebe. Fisher kernel based relevance feedback for multimodal video retrieval. In *ICMR*, 2013.

[128] Linjun Yang and Alan Hanjalic. Supervised reranking for web image search. In *MM*, 2010.

[129] David Grangier and Samy Bengio. A discriminative kernel-based approach to rank images from text queries. *PAMI*, 30(8):1371–1384, 2008.

[130] Winston H Hsu, Lyndon S Kennedy, and Shih-Fu Chang. Video search reranking through random walk over document-level context graph. In *MM*, 2007.

[131] Liqiang Nie, Shuicheng Yan, Meng Wang, Richang Hong, and Tat-Seng Chua. Harvesting visual concepts for image search with complex queries. In *MM*, 2012.

[132] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *MM*, 2005.

[133] Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007.

[134] Stephen Poythress Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[135] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[136] Michel Berkelaar. lpsolve: Interface to lp solve v.5.5 to solve linear/integer programs. *R package version*, 5(4), 2008.

[137] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[138] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[139] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *PAMI*, 34(3):480–492, 2012.

[140] Faisal Khan, Bilge Mutlu, and Xiaojin Zhu. How do humans teach: On curriculum learning and teaching dimension. In *NIPS*, 2011.

[141] Sumit Basu and Janara Christensen. Teaching classification boundaries to humans. In *AAAI*, 2013.

[142] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei, and Daphne Koller. Shifting weight-s: Adapting object detectors from image to video. In *NIPS*, 2012.

[143] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *PAMI*, 35(8):1798–1828, 2013.

[144] Yoshua Bengio. Evolving culture versus local minima. In *Growing Adaptive Machines*, pages 109–138. Springer, 2014.

[145] Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Baby steps: How less is more in unsupervised dependency parsing. In *NIPS*, 2009.

[146] Ye Tang, Yu-Bin Yang, and Yang Gao. Self-paced dictionary learning for image classification. In *MM*, 2012.

[147] J. Supančič III and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, 2013.

[148] M. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *ICCV*, 2011.

[149] T Mitchell, W Cohen, E Hruschka, P Talukdar, J Betteridge, A Carlson, B Dalvi, M Gardner, B Kisiel, J Krishnamurthy, et al. Never-ending learning. In *AAAI*, 2015.

[150] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arX-iv:1406.2080*, 2014.

[151] Xiaodan Liang, Si Liu, Yunchao Wei, Luoqi Liu, Liang Lin, and Shuicheng Yan. Towards computational baby learning: A weakly-supervised approach for object detection. In *ICCV*, 2015.

[152] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[153] Deyu Meng and Qian Zhao. What objective does self-paced learning indeed optimize? *arXiv preprint arXiv:1511.06049*, 2015.

[154] Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *JMLR*, 11:1081–1107, 2010.

[155] Pinghua Gong, Changshui Zhang, Zhaosong Lu, Jianhua Z Huang, and Jieping Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML*, 2013.

[156] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, pages 894–942, 2010.

[157] Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[158] Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *NIPS*, 2013.

[159] Shankar Vembu and Sandra Zilles. Interactive learning from multiple noisy labels. In *ECML*, 2016.

[160] Maciej Zieba, Jakub M Tomczak, and Jerzy Swiatek. Self-paced learning for imbalanced data. In *Asian Conference on Intelligent Information and Database Systems*, 2016.

[161] Enver Sangineto, Moin Nabi, Dubravko Culibrk, and Nicu Sebe. Self paced deep learning for weakly supervised object detection. *arXiv preprint arXiv:1605.07651*, 2016.

[162] Jian Liang, Zhihang Li, Dong Cao, Ran He, and Jingdong Wang. Self-paced cross-modal subspace matching. In *SIGIR*, 2016.

[163] Dan Xu, Xavier Alameda-Pineda, Jingkuan Song, Elisa Ricci, and Nicu Sebe. Academic coupled dictionary learning for sketch-based image retrieval. In *MM*, 2016.

[164] Dan Xu, Jingkuan Song, Xavier Alameda-Pineda, Elisa Ricci, and Nicu Sebe. Multi-paced dictionary learning for cross-domain retrieval and recognition. In *ICPR*, 2016.

[165] Dingwen Zhang, Deyu Meng, Chao Li, Lu Jiang, Qian Zhao, and Junwei Han. A self-paced multiple-instance learning framework for co-saliency detection. In *ICCV*, 2015.

[166] Qian Zhao, Deyu Meng, Lu Jiang, Qi Xie, Zongben Xu, and Alexander G Hauptmann. Self-paced learning for matrix factorization. In *AAAI*, 2015.

[167] Hao Li, Maoguo Gong, Deyu Meng, and Qiguang Miao. Multi-objective self-paced learning. In *AAAI*, 2016.

[168] Changsheng Li, Fan Wei, Junchi Yan, Weishan Dong, Qingshan Liu, and Hongyuan Zha. Self-paced multi-task learning. *arXiv preprint arXiv:1604.01474*, 2016.

[169] Yulia Tsvetkov. *Linguistic Knowledge in Data-Driven Natural Language Processing*. PhD thesis, Carnegie Mellon University, 2016.

[170] Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*, 2016.

[171] Jie Fu, Zichuan Lin, Miao Liu, Nicholas Leonard, Jiashi Feng, and Tat-Seng Chua. Deep q-networks for accelerating the training of deep neural networks. *arXiv preprint arXiv:1606.01467*, 2016.

[172] Yanbo Fan, Ran He, Jian Liang, and Bao-Gang Hu. Self-paced learning: an implicit regularization perspectiv. *AAAI*, 2017.

[173] Agata Lapedriza, Hamed Pirsiavash, Zoya Bylinskii, and Antonio Torralba. Are all training examples equally valuable? *arXiv preprint arXiv:1311.6510*, 2013.

[174] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[175] Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[176] Zhenzhong Lan, Xuanchong Li, and Alexandar G Hauptmann. Temporal extension of scale pyramid and spatial pyramid matching for action recognition. *arXiv preprint arXiv:1408.7071*, 2014.

[177] Eleonora Vig, Michael Dorr, and David Cox. Space-variant descriptor sampling for action recognition based on saliency and eye movements. In *ECCV*, 2012.

[178] William Brendel and Sinisa Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011.

[179] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu, and Chong-Wah Ngo. Trajectory-based modeling of human actions with motion reference points. In *ECCV*, 2012.

[180] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Recognizing activities with cluster-trees of tracklets. In *BMVC*, 2012.

[181] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *arXiv preprint arXiv:1502.07209*, 2015.

[182] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[183] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[184] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

[185] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3): 59–79, 2010.

[186] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3): 475–494, 2001.

[187] James E Falk and Karla L Hoffman. Concave minimization via collapsing polytopes. *Operations Research*, 34(6):919–929, 1986.