A Guided, Low-Latency, and Relevance Propagation Framework for Interactive Multimedia Search

Eric Zavesky

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2010

ABSTRACT

A Guided, Low-Latency, and Relevance Propagation Framework for Interactive Multimedia Search

Eric Zavesky

This thesis investigates a number of problems associated with the efficient and engaging ways of executing a multi-level interactive multimedia search. These problems are of interest as the availability of multimedia sources, both professional and personal, continues to grow in tandeom with the need for users to search these libraries for consumable entertainment, captured personal memories, and automatically events with little or no forethought to manual indexing.

Multimedia search refers to the retrieval of relevant content from databases containing multimedia documents. Interactive search means that a user is exploring a dynamic set of results according to parameters that he or she explicitly chose in response to a specific search topic. Multi-level search is the full utilization of a user's interaction with a system to not only provide explicitly requested results, but also to observe a user's preferences and implicitly personalize those results with only interactions the user has already performed. The goal of this thesis is to develop a framework that both guides the user through his or her search process by providing dynamic suggestions and information from automatic algorithms while simultaneously leveraging cues observed during the search process to provide a customized set of results that most precisely matches the user's search target. Upon achieving this goal, the system is aiding the user through both explicit interaction and subsequent result personalization from implicit search choices. A prototype of the proposed system, called *CuZero*, has been implemented and evaluated across multiple challenging databases to discover new search techniques previously unavailable.

Addressing problems in traditional query formulation, a system that interactively guides the user is proposed. While previous works allow a user to specify different modalities for

a multimedia search like textual keywords and image examples, this work also introduces a large library of 374 semantic concepts. Semantic concepts use pre-trained visual models to bridge the gap in perception between what a machine computes for a multimedia document and what a user can do with that computation. For example, a user need only utilize the concept "crowd" to return content containing large numbers of people attending a basketball tournament, a political protest, or an exclusive fashion show. Building on the familiar technique of text entry (typing in text keywords), the system returns a small subset of dynamically suggested concepts from a lexical mapping and statistical expansion of the user's entered text. These suggestions both engage and inform the user about what the system has indexed with respect to the current query text. Additionally, the introduction of a unique query visualization panel allows the user to interactively include arbitrary modalities (text, images, concepts, etc.) in his or her query. Traditional trial-and-error search with these different query parameters is avoided because the system allows the user to visually arrange his or her query according to personal intuitions about the search topic. Finally, while formulating the query, time otherwise lost while the user is thinking is utilized simultaneously evaluate and load results for the current query at-hand.

After a query is formulated during a guided and informative process, the formulation panel is subsequently utilized for query navigation, allowing the user to instantly review numerous query permutations with no perceived latency. With the intuitive mantra "closer to something is more like it", the user is prepared to instantly change the weights of the various parameters in his or her query. To accommodate this flexibility, previous systems in interactive search resorted to burdening the user with a secondary query specification stage to tweak individual modality weights. However, the proposed approach to result browsing allows the user to navigate the query and result space in parallel, spanning a wide breadth of query permutations or a deep result depth for any one query permutation. Another classic barrier in multimedia search is the sensible inclusion of new search modalities; if no longer constrained to color or text cues, how can one include motion, audio, and local object similarity that has no textual correspondence? Fortunately, the proposed query navigation panel was created in such a way that any modalities developed in the future can be included with no additional algorithmic changes. This flexibility is best exemplified during the result

browsing process, where a user can include another image for example-based search or a personalized snapshot of seen results into the query to quickly hone in on desirable results.

A final proposal in this work is a scalable and real-time result personalization technique. One of the fastest ways to help a search system identify results relevant to a user's search topic is to explicitly solicit a user's preference. With interactive systems, this usually means interrupting the result browsing process and asking the user whether they like a particular result. While numerous methods for this type of relevance feedback have been proposed, they all currently trade performance for speed. Typically this problem is due to the scale of the database in question and the number of results required for a system to eliminate confusion about the user's search target. On one end of this spectrum, supervised learning techniques can scale to process millions of results but also require a large number of labels. On the other end, graph-based semi-supervised techniques have been able to achieve promising performance with only a handful of labels, but the time to construct a graph is unacceptable for real-time scenarios. In this work, state-of-the-art graph-based label propagation is aided by data approximation techniques, in a proposed algorithm that is able to achieve higher accuracy in only a small fraction of the computation time when evaluated on a standard benchmark dataset. Using the real-time implementation of this technique, user search results can be personalized without the need to solicit result preferences en mass.

The specific contributions of this thesis are as follows. (1) A new system for query formulation, traditionally relegated to static, non-informative interfaces, is proposed that keeps the user engaged in the process and dynamically proposes query suggestions based on knowledge from the system. (2) A technique for visual query space navigation is proposed that allows an intuitive exploration of several query permutations with no additional latency from the explicit, real-time manipulation of modality weights. (3) Utilizing a proposed hybrid of graph-based label propagation and data approximation techniques, user search results are personalized in real-time using implicit user preferences. These proposals are included in a prototype system called *CuZero* that is evaluated to produce unique search opportunities unavailable to existing multimodal search systems.

Table of Contents

List of Figures

List of Tables

Acknowledgments

During the past six years at Columbia, I have enjoyed the company of several colleagues, mentors, and friends that jointly encouraged me to explore exciting new topics of technical merit and ultimately complete the work that embodies this thesis. While this work was far from trivial, the advice, guidance, and technical insight of these people made the experience an enjoyable one.

I offer thanks to my advisor, Shih-Fu Chang, for his guidance and willingness to take a chance on an overly optimistic graduate student and eventually sponsor me as an advisee. I am consistently amazed at his ability to balance an impeccable technical background, responsibilities to students as an advisor, and an ever-expanding knowledge of the field as both an observer and leader. I am also appreciative of his ability to listen to ideas in their infancy and amplify them with years of experience into concepts that not only match the state-of-the-art but seek to surpass it.

I offer my gratitude to the whole of my thesis panel, Dan Ellis, Rui Castro, Paul Natsev, and Behzad Shahrary, who graciously acknowledged the merit of this thesis but pushed me to extoll additional technical achievements that I had not fully realized in its initial writing.

I am appreciative of the TRECVID community at large and the staff members of NIST who tirelessly prepare, disseminate, and evaluate several multimedia tasks every year. Without their efforts, this thesis would be entirely hypothetical instead of rigorously evaluated with internationally accepted benchmarks.

I would also like to thank my current and former peers from Columbia: Lyndon Kennedy, Wei Jiang, Akira Yanagawa, Jessie Hsu, Michael Mandel, Wei Liu, Yu-Gang Jiang, Lexing Xie, Tian-tsong Ng, Kevin Jamison, Winston Hsu, Junfeng He, and Jun Wang. These individuals not only offered their own technical twists on problems that I encountered, but they also enriched my life with their own areas of expertise and culture. Through

participations in critical group and individual discussions alike, I now strive to contribute to our community in large technical leaps instead of incremental steps.

Thanks to several members of AT&T Labs, Zhu Liu, Behzad Shahrary, and Dave Gibbon, who provided numerous internship opportunities in one of the few remaining industrial research labs in the world. During these intervals in time, they embraced my desire for real-world applications of the research in this thesis while fostering opportunities for long-term collaborations.

For the encouragement and opportunities for reflection, I must also acknowledge my friends Kevin Ludlow, Dave Guezuraga, Michael Crockett, Dreux LaViolette, and David Gratz. Although our backgrounds and stations in life may differ, you provided unending comments of excitement, confidence, and the occasional joke around a campfire.

Finally, my wife, Amy, my parents, Mary and Byron, and my brother, Alex, deserve mention here for the collective patience, persistence, and caring that brought me to the happy destination of today. However, instead a written thanks in this paragraph, I chose to repay these deeds in kind for decades to come.

Dedicated to my family:
those who are currently leaving their mark on the world
and those who already have.

Chapter 1

Introduction

This thesis proposes a system framework incorporating several new techniques that bridge the gap between user interaction and system information capabilities and facilitating the fast and precise search of multimedia datasets.

1.1 Background

The explosive growth of storage devices and new ways to consume multimedia content have produced unique opportunities for individuals and businesses alike to both create distinct video content and consume content from all corners of the earth. The precipitous decline in price and increased availability of capture devices (personal cameras and mobile phones), storage media (traditional hard-disk and optical media), and high-speed internet connectivity has allowed users to share and consume both personal and produced multimedia content. Looking at traditional television consumption, at the end of 2008, the average viewer watches more than 151 hours per month while those viewers using personal video recorders (PVRs) watch over 7 hours per month, an increase of 33% from 2007 [83]. This high consumption rate and growth of non-traditional use indicates that while the average consumer watches a lot of content, these consumers are increasingly adopting new technologies that help to capture and browse content specific to their preferences. Looking at personal content, the number and size of digital photos, which are now digitally retained for a lifetime, is also so large that additional assistance in managing this content is needed;

a single camera can now store up to 16 gigabytes of content (up to 120k images) at resolutions up to 10 mega-pixels. Finally, as the sheer volume of available content grows, the pervasiveness of mobile and streaming solutions are also changing ways that people consume content. Online and mobile consumption per month is 2.88 hours and 3.7 hours respectively and the combined number of online streams consumed in single a month has grown to over 11 million, increasing 24.8% between 2008 and 2009 [84].

Although large amounts of multimedia content are increasingly produced and consumed, a large portion of available content often goes unused. Users want to find archived recordings of favorite television shows, amusing web clips, or even personal home videos but they are troubled by the difficulty of finding these videos. This trouble often comes from two underlying challenges: how to index, or organize, and search the videos in an intuitive and logical way (i.e. how to search and navigate results) and how to customize each search for the specific needs of each user (i.e. specific identification of a user's search target).

Looking at answers to the first problem of indexing and search, numerous algorithms have been developed to help break apart and analyze new video content. For example, machines can automatically analyze and segment a produced video (i.e. a news broadcast or television show) and achieve a 97% agreement with segments that a human would generate [72]. Similarly, owners of modern digital cameras often enjoy the ability to detect faces in real-time. This technology, which allows cameras to detect frontal faces for better focus control, ideal lighting conditions for a picture, and even "closed-eye" conditions, was first developed as a generic algorithm [36], adapted for face detection [118], and proposed as a method for automatic construction of custom photo albums [23]. While both tasks may be slightly more difficult for surveillance footage or user-generated content, like internet clips or home videos, solutions in these research areas are largely more mature than other components. A second example that both indexes and searches for multimedia content is currently quite topical. Algorithms that allow the identification of near-duplicate content are increasing in popularity and utility with mobile devices. Near-duplicate audio is heard every time a radio station plays a new artists hit song and can be identified with only a few moments of a song [120], near-duplicate video occurs every time a single event or location is captured by multiple people with differing viewpoints like a politician's public

address [137],[21] or a music concert [59]. Near-duplicate identification is a specific type of a multimedia search that has received a great deal of focus because of its applications in consumer technology and content copyright identification. While these applications are examples of increasing success, the larger problem of organizing and searching produced (i.e. archives of all public broadcasts) and personal content (i.e. all personal photos, home videos, etc) libraries goes largely unanswered. Currently, the state-of-the-art for this field can be boiled down to either organization by manually specified tags [124] or basic time-date-location organization as verified in responses to consumer surveys [37].

Contrary to the proliferation of advanced machine representations of multimedia, the second problem of customized search has not been as widely studied, leading to a lack of work that directly tackles the interaction of a user and a search interface. Ideally, an interactive search system would simultaneously harness both the explicit user actions with the system and the implicit user preferences available from user navigation and search choices. Unfortunately, due to complexity of scale or the pace of innovation, interactive systems for multimedia search are largely available only in academic or industrial research environments and often place less emphasis on learning long-term user preferences. Acknowledging that fact, a number of systems that have analyzed tactile (and large scale) interaction during search [66], collaborative environments where two or more users alternate interaction roles [89], and even systems that enforce high-speed result inspection [46] and highly dynamic interfaces [125] have been developed. Each of these systems explores a new technique in interaction, but by doing so they also decrease intuitiveness for novice users. Web search engines reside in the middle of this continuum and are hybrid systems that observe both explicit user activity and implicit user preferences. In a very unique role, Web search engines often present a very simple and intuitive interface: a text entry region, a "search" button, and links that allow users to jump to different result pages. In this interface, user actions are most related to clicks for relevant results or navigation difficulties; if a user wants to exclude or hone-in on certain results he or she must interrupt the search process and explicitly teach the search system these rules. Recommendation engines reside at the other end of the spectrum that focus almost exclusively on implicit user preferences for a personalized search experience. An active field of research, recommendation engines (like

those for movies [10] and music [76]) harvest information from a large number of user responses in hopes of discovering agreement between users to collaboratively recommend new results. One weakness of these recommendation engines is that they are largely relegated to off-line processes where large resource requirements prohibit the search engine from immediately customizing a user's search. In summary, local solutions to customized search exist for many special circumstances, but few have been able to encompass all forms of user interaction.

The problems discussed here exist for all types of multimedia data from text-only books or Web pages to rich multimedia videos that are pre-produced or distributed in real-time from personal cameras. While existing search engines continue to grow in algorithm complexity and quantity of analyzed content, a sensible solution for interactive search is still unavailable. Fortunately, in the same way that online text search engines revolutionized information discovery from anywhere in the world, new techniques for multi-level interactive search utilizes both explicit user interactions and implicit user preferences.

1.2 Motivation

In order to answer all search needs for users in a natural and intuitive fashion, it is necessary to combine lessons from many of the modern solutions from the previous section. First, a generic system that provides multimedia search with any arbitrary representation should be created. For example, a user should be allowed to search his or her personal video archive by date, textual tags, or even with examples of specific people or scenes that are important. Second, this system should not blindly execute a search when requested, but instead allow a user to interactively discover better search strategies as they explore the available results. Third, as a user engages the system through a single search session, or through several search sessions, the system should automatically customize the user's search to provide highly-related results and minimize the number of system-required interruptions that divert a user from the primary search task at hand.

Figure 1.1 illustrates the overall flow of a typical multimedia search engine. With the automated indexing of multimedia sources, interactive search is performed by first formulating

a query, inspecting results, and reranking those results for a personalized search experience. The remainder of this section discusses these three interactive search steps and poses motivating questions about how they can each be improved to satisfy the requirements defined above.

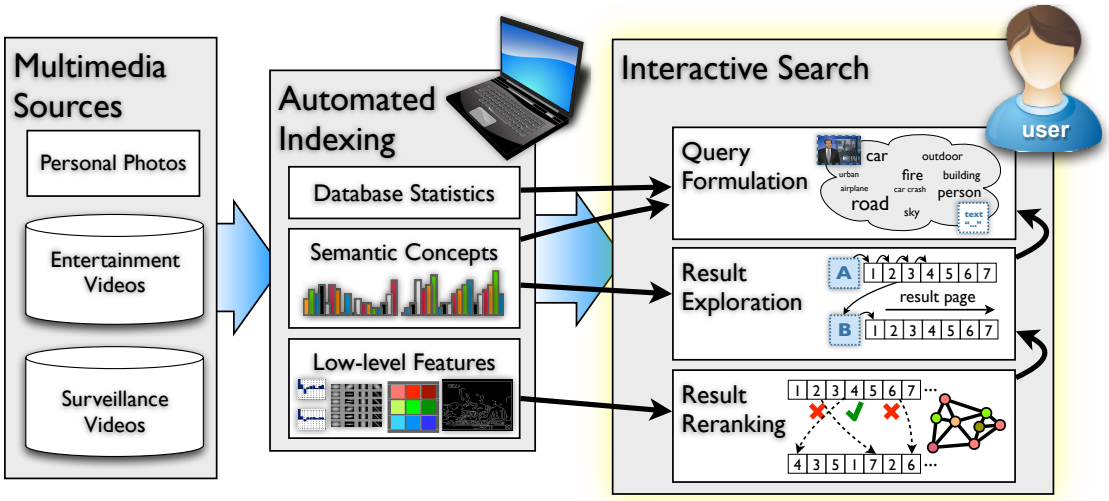


Figure 1.1: A typical multimedia indexing and search system. This work is focused on introducing intuitive and powerful techniques that naturally couple machine and human actions during interactive search.

Query formulation is the process of describing the tools necessary to answer a search topic. For example, a well-known method of query formulation for searching the Web is to type a few keywords into a text entry box. While quite intuitive because a user specifies the exact keywords that he or she is interested in finding, this method is also extremely simple and can lead to problems in ambiguity. One counter-example presenting is the search topic "cruise in manhattan". With text alone, a system (or human) can identify "manhattan" as a borough in New York City, but the same system is left to guess about references to *cruise*, which may refer to the action of driving, an event involving a riverboat, or a person like Tom Cruise. Is it possible that the user could help the system with example images, a specific date range that may help to reduce acceptable answers, or even a description of the desired scene context? Expanding on the last question, if systems are to obey the requirement of intuitiveness, are there acceptable generic descriptions of a scene that both

novice and expert users could agree upon? Specifically, is there a way for the user to introduce keywords relating to the content of the image, instead of events or spoken words, like "grass" or "water" that have a well-known visual interpretation yet are generic enough to represent many different views? During query formulation, the user enters keywords or submits image examples with no feedback from the system. While it is increasingly common for a Web search engine to offer keyword suggestions based on other user queries, what happens if a user's specific search target is not available? The most common answer is that the user must simply try the new keyword combination and see if any of the results are correct, but surely there are better solutions.

Result inspection is one of the most under-utilized forms of user interaction in search systems. Currently, after formulating a query and executing its search an ordered list of results is returned. While this list may have some machine-computed indicator of relevance (i.e. a percentage or rating), the burden is now on the user to traverse the list and make guesses about parts of the query did and did not work. After this non-trivial guessing process, the user must then return to query formulation, revise his or her query, and restart result inspection. This lack of system interaction forces the user to ask two questions: what did the search engine try to find and what are my alternatives. Given that the system knows the user's query before each search, is there not a more intuitive way of organizing and presenting results to the user?

Result personalization is a process that dynamically reranks (compute customized scores and reorders) search results to better fit a user's current search requirements. While personalization can be either long- or short-term corresponding to user requirements from multiple search sessions or only one, a sensible solution must first be proposed. Existing systems offer the user two unfavorable options that trade-off reranking quality for speed. High-quality reranking often requires the user to interrupt the search process and explicitly refine his or her search with new query criterion, like keywords or content examples. High-speed reranking can produce new results, but the results can quickly drift away from the original query and search target. Analyzing new reranking techniques from other fields, are there any solutions that both scale for large problems and produce high-quality results with only implicit user activity and not additional query modifications?

1.3 Overview of Approaches and Contributions

The main contributions of this thesis are as follows.

- **Explicit Guided Query Formulation** A new form of guided system interaction is proposed to best inform users during the query formulation process. Leveraging hundreds of pre-trained visual concepts [129] and several automated suggestion techniques, the user works with the system to formulate the most relevant and helpful query based on video content, not just text or other metadata. The suggestion techniques range vary from lexical approaches to statistical data mining to propose a small subset of visual concepts that the user might have overlooked or been unaware of. During query formulation, a number of speed and resource optimizations are also proposed to reduce the maximum perceived search time to an average of five seconds and keep the user fully engaged in a single, well-informed query formulation session with nearly instant response times [134].

- **Explicit Real-time Query-space Navigation** An intuitive query visualization and navigation system is proposed that traverses not only the *result space* but also the *query space* simultaneously in such a way that multiple permutations of a user's query can be evaluated in real-time. After search execution, users are immersed in results to be inspected. Existing solutions lock a user into a tiresome linear inspection of result or alternatively jump to tangentially related results, which may further confuse the user. The unique availability of the proposed real-time interaction system allows the user to break from trial and error search behaviors and more quickly identify what parts of a query help and how to improve that query with alternative text, concept, or image query parts. Maximizing the intuitiveness and ease of query manipulation, a prototype system called *CuZero* was evaluated with a number of diverse search topics across different datasets and multiple examples of usages not possible with other existing works are discussed. Finally, observing the pervasiveness of mobile technology, a number of additional speed and application discussion are included to demonstrate the flexibility of the system and its viability in distributed but collaborative multimedia search.

- **Automatic Result Reranking** A new method is proposed that utilizes user labels to automatically rerank search results with very high accuracy. Result reranking is a technique that uses automated algorithms to leverage user preferences (i.e. "I like this" or "I don't like this" labels) to personalize his or her search results. Prior works in the field often select a subset set of all results then require the user to reformulate his or her query. Other works achieved high accuracy for result reranking but either required exponentially growing computation times or broke down when the result set grew too large. In the proposed method, labels over an implicitly defined result set and data reduction techniques are used so that the user is neither derailed from the search task at-hand nor asked to wait lengthy periods of time for reanking processes to execute. This technique is evaluated on two tasks: concept classification of the TRECVID2008 development dataset and multi-class classification of the USPS digit dataset. In the realistic TRECVID dataset, a mean average precision (MAP) of 0.18 over twenty concepts is an absolute gain of 350% over an initial nearest neighbor query. In the traditional but challenging USPS dataset, classification of ten handwritten digits achieves an error of only 16%. In both cases the proposed approaches can execute in less than one-sixth of the time required when compared to most state-of-the-art systems. Also using the USPS dataset, additional analysis is conducted demonstrating the robustness of the proposed method as compared to these state-of-the-art systems and as the number of results grows beyond previous computational limits.

1.4 Organization

The remainder of this thesis is organized as follows. In chapter 2, the fundamentals of analyzing a multimedia document and indexing it by its machine-computed descriptions is provided. This chapter also includes the definition of critical representations and introduces the idea of *semantic concepts* and multi-modal search. Incorporating automated indexing and search tools, chapter 3 reviews the components of an interactive search system. This includes techniques available for users to create queries from search topics, perform mapping

of keywords into the visual domain, interactively navigate search results, and even indicate good and bad answers for the search topic with *relevance feedback*. With this working knowledge, open problems for interactive search and feedback systems are also presented.

Chapter 4 directly tackles two open problems in interactive search: query formulation and result browsing. By providing a system that informs the user of its capabilities and a user's alternative query options, a rich query formulation experience preempts user frustration and bad search habits that may otherwise surface. Result browsing, although seemingly a simple process, is transformed from a laborious list-based memory game to one that encourages rapid exploration of many query possibilities that continue to help the user understand what it knows about the searched content. In this chapter, a prototype system abiding by these proposals is also presented and evaluated on a number of diverse and challenging datasets.

Acknowledging that search results in an interactive system are seldom perfect, chapter 5 closely investigates the topic of result reranking. A unique formulation using dataset exemplars for approximation is combined with high-precision, graph-based label propagation to create a semi-supervised machine learning algorithm that scales to large datasets, requires only a few user labels, and can be executed in real-time scenarios.

Finally, chapter 6 summarizes the thesis and discusses both conclusions and opportunities for future work in the presented areas.

Chapter 2

Review: Content-Based Video Search

Content-based search, much like other search tasks, requires strong underlying machine representations for a multimedia document and efficient methods for organizing those representations. In this chapter, section 2.1 introduces methods for automatically segmenting, analyzing, and indexing a video with a discussion of both traditional and state-of-the-art representations for video. After indexing, section 2.2 introduces methods for searching directly with video via speech or image content and indirectly by visual concept representations and section 2.4 demonstrates how multiple representations can be used in various multimodal search scenarios. Finally, section 2.5 summarizes these foundations and reviews potential pitfalls from fully automated search.

2.1 Content-Based Multimedia Indexing

Content-based multimedia indexing is a broad description of the process to organize a *dataset* (or set of documents) by their underlying components. In multimedia search, a document is composed not only from text keywords (as are Web documents), but also from rich audio and video signals. A single audio or video signal can have multiple *features*, or descriptions of its parts, where each feature has a representation that a machine can compute and a representation tangible to humans. For example, in content-based multimedia

indexing, both the color "green" and the spoken name "Barak Obama" can be represented by canonical features that have distinct visual and aural meanings for both human and machine.

Unfortunately, from the origins of image retrieval and video retrieval to modern day research, no work in the field has been able to fully close the sensory and semantic gaps in multimedia search [102], [24]. The *sensory gap* is the difference between live perception in the world and a computational (machine-based) recording. In this context, sensory gaps can not be overcome with current computer-based interactions because information loss occurred when the original video signals were captured. The *semantic gap* is the difference in information that a machine can extract versus the interpretation of that information by a person. The semantic gap, also a challenge posed by computational limitations, can be partially bridged by replacing the information lost in user perception with piece-wise approximations, as discussed in section 2.1.3. With this representation of multimedia content, flawed as it may be, the remainder of this section focuses on the discussion of the indexing pipeline illustrated in figure 2.1. This figure depicts the segmentation of video content into atomic units of video and audio signals, the derivation computation of viable multimedia features, and the use of these features to partially bridge bridge the semantic gap with a large collection of visual semantic concepts.

2.1.1 Content Segmentation

Contrary to static images or text-documents, videos also posses an additional dimension of time. Naturally, the complexity of video content has given rise to two types of segmentation: those that preserve syntax structure and those that preserve semantic structure. Within video, syntax refers to the structure of its parts (i.e. camera, speaker, or scene changes) or whereas semantics refers to the coherence of the content itself (i.e. a movie dialog about a character or a single story in a news broadcast). Even today, large disagreements on semantic structure can exist between humans so a segmentation task may be unreasonable for a machine. Segmentation with syntactic structure, however, is largely agreed upon and often pre-defined for highly produced content, like broadcast news and cinema. Fortunately, stemming from long-standing research in the field, a level of 97% human-machine agreement

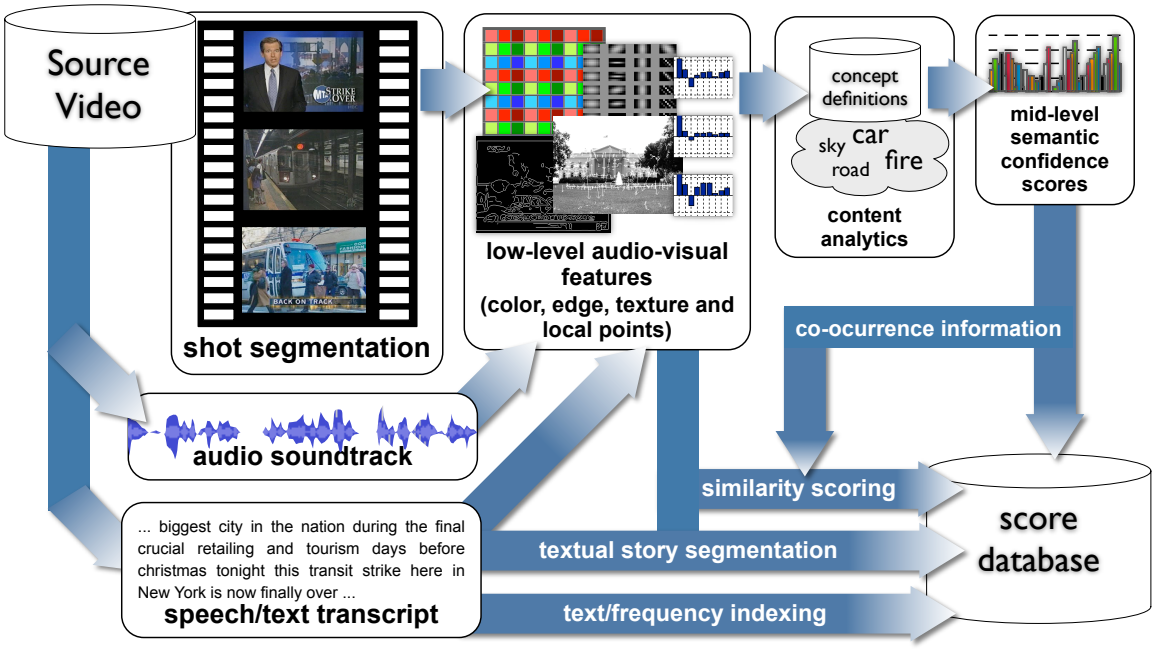


Figure 2.1: The multimedia indexing pipeline: a video is segmented, features extracted, and is scored by semantic concepts during indexing.

can now be achieved with automated techniques for syntax-based segmentation [72]. Recognizing the importance of semantic analysis, information lost during initial segmentation is recovered through alternative approaches discussed in section 2.1.3. Although this thesis experiments mostly with video-based documents, the term *clip* is used to represent content that may be either video- or audio-based in nature.

Clip segmentation can be performed either at fixed intervals (i.e. every ten seconds) or at intervals that preserve the syntax of content, but both share the goal of obtaining a *shot*, or small temporal segment, of content that is highly cohesive in its semantics. The specifics of shot segmentation algorithms and their relative performance on a closed set of datasets are omitted from this thesis but are available in other overviews of published work [100]. In this thesis, smaller segments of shots called *subshots* are computed, which are generally two seconds or less in duration. Subshots are the smallest time-unit of analysis are thus represented by a single keyframe. Subshots are a better basis for analysis because they offer a very temporally short, visually coherent segment of video, which may accidentally be spanned in longer shot-based segments. To avoid problems related to erroneous segmenta-

tions, like those caused by camera flashes, advanced graphical transitions, or fades to or from black, a subshot's keyframe time is derived from the temporal center of a subshot. After subshot segmentation, low-level features and mid-level semantic concept scores (discussed below) and then higher-order statistics can be automatically computed for a clip. Here, textual content in the document is stemmed and enumerated, similarity between images is computed, and the frequency and simultaneous appearance (co-occurrence) of concepts are indexed in a multimedia database for future retrieval during a user search. Content analysis at the subshot level (instead of the shot or clip level) permits semantic description at a very granular level and increases the chances of finding near-duplicate content from other subshots. While subshots are ideal for indexing, they are often too short for humans to clearly understand the event or activity that is occurring in a single subshot so during result review shot-level video segments are utilized.

2.1.2 Low-level Features

After subshot segmentation and keyframe extraction, the video and audio signals in subshots can be analyzed by specific algorithms to extract low-level features. For any given audio or video signal, aligning the perception of the machine to the perception of a human is a non-trivial task, embodied by the semantic gap. Low-level features, although not a full solution allow machines to encode descriptions of a subshot's video and audio signals for subsequent processing and are commonly grouped into four general types: metadata, text, audio, and video.

Metadata is numerical or textual information that describes macro-level properties of a piece of content. Examples of metadata in Web pages is the title, publication date, or author. Similarly, personal photos from a digital camera are also described by meta-data like the capture date and author and with the proliferation of GPS hardware, cameras are increasingly encoding each image with the geo-spatial position of the camera at the time of capture. Metadata is generally too vague to search with alone, but in the presence of other meaningful features, it can be used to quickly filter or reduce the number of results that must be inspected after a search, as exemplified in section 2.4.

Textual low-level features are most commonly frequency counts of one or more adjacent

words (n-grams) from a fixed dictionary. This dictionary can include the words themselves, proper nouns (names), and even the part-of-speech of a word (subject, verb, etc). To reduce the number of words in a dictionary, a process called stemming is applied, which produces a common etymological root (i.e. smile from smiling, smiled, smiles, etc.) [90]. Although these features are derived from words, they can be extracted from transcripts of automatic speech recognition (ASR), machine translations (MT), closed-caption data (CC) [96], optical character recognition (OCR) [95], and even high-quality manual transcriptions [40]. Text features are the *de facto standard* for online Web search engines, but when applied without further refinement to video or audio search problems in semantics (will a speaker explicitly say "sunset" when reporting on a story about the outdoors?) and syntactic (a financial report about the manufacturer "Toyota" may never contain an image of a car).

Audio low-level features are most commonly associated with speech, such as pitch (formant frequency) and prosody (rhythm and pitch) of a spoken dialogs, but music and environmental sounds can also be represented with frequency-domain coefficients like Mel-frequency Cepstral Coefficients (MFCC's) [77]. For video search, audio features have been used in concert with visual features for enhanced story segmentation [51]. The use of audio features for search is limited in this thesis, but their success in audio-based duplicate search [120] and recommendation engines [76] has shown promising gains in recent work.

While the computational description of a visual signal is still an area of research, visual features can be used to capture both global (i.e. whole image) and local region (i.e. 200 pixel) signal representations. This work focuses on multimedia indexing with the select set of visual features briefly described below.

- **Grid Color Moments** describe the mean, variance, and skewness of pixel intensities within an image region [108]. Grid color moments analyze an entire image over a regular grid and compute three moments for each color channel. Color moments (tempered with an appropriate color space, like LUV) are not as sensitive to illumination shifts, as long as that shift occurs with a non-chromatic light source. The grid-based analysis adds a geometric requirement for matching and thereby assures that visual signals in a scene (i.e. the yellow glow of a sunset over the green grass of a field) are not dramatically transposed. Another popular color representation is the

color opponent model [39], which adds additional invariance to global shifts in color because it captures the relative difference of color channels and has been shown to work particularly well in near-duplicate scenarios. In this work, images are analyzed via the color grid moments approach in the LUV color channel with a 5x5 region grid.

- **Gabor Texture** features are computed from a multi-scale, multi-direction application of 2D pixel intensity filters [116]. This method for texture approximation was shown to be more robust than applying template matching with strict pixel-based patterns because it does not require the pre-computation of a large texture library but it can still match a diverse set of low frequency (i.e. clouds, rolling hills, etc.) and high frequency (i.e. buildings, logos, etc.). In this work, the filters are applied at each combination of 8-orientation and 6-scale settings to capture the statistics of patches.

- **Edge Direction Histograms** are computed in a two stage process: first derive the edge map for an image (i.e. via Sobel processing) and then accumulate pixel edge orientations into a histogram defined by regularly spaced angle intervals [3]. The edge direction histogram allows for some geometric variance in a visual signal that may have been excluded by color or texture features (i.e. different patterns from multiple brick courses, or rows, in a building). This work uses quantizes directional bins at five degrees and one empty bin for empty pixels.

- **SIFT bag-of-features** (BoW) harness SIFT or scale-invariant feature transform, that was originally created to ease the recognition of scenes in computer vision [74]. In a SIFT BoW representation, a three step process is executed for each image. First, interest point detection attempts to find points of interest by approximating the human vision system with corner detectors (Harris affine [78]) or multi-scalar gradient (difference of Gaussian [74]) points. Next, interest point descriptors from a grided region around each point are computed with a local gradient operator and a histogram captures the normalized orientations of these gradients. Finally, the entire SIFT vector is mapped into a visual codebook (or dictionary) utilizing either the soft- [54] or quantized-weights [88] technique. In this work, a codebook has 500 soft-weighted entries derived from DoG (difference of Gaussian) interest point detection. SIFT BoW

representations typically use local features computed from grayscale images. Recent work has shown that the incorporation of color channels and different interest point detectors can further improve matching performance [117].

Low-level features are often indexed directly in search engines; for every subshot in the database, low-level vectors are computed and stored. The efficient indexing of these low-level features is an open challenge because each additional subshot in a dataset requires a number of different low-level features. Also, compounding the problem of efficient indexing, two large perceptual gaps exist between low-level machine representations and human understanding so more natural representations must also be explored.

2.1.3 Mid-level Semantic Concepts

Looking back to the *semantic gap*, one major struggle for all users of a multimedia indexing system is understanding how a machine representation corresponds to the content of a clip. Illustrated in figure 2.2, three conceptual levels of features are generally discussed: low-level features, mid-level semantics, and high-level descriptions. In this example a subshot from a news broadcast about soccer is analyzed. A mid-level representation adhering to one or more

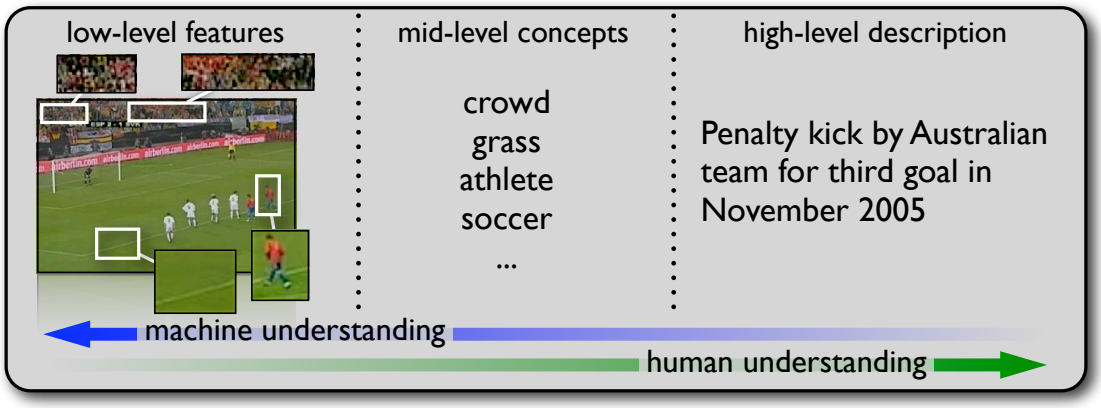


Figure 2.2: A contrast of machine and human understandings of multimedia content using visual low-level features, mid-level semantic concepts, and high-level textual descriptions as example representations.

physically describable ideas is called a *semantic concept*. The high-level information about the specific scoring event or team involved is lost in the mid-level representation. However,

mid-level descriptions of the content like "crowd" or "athletes" that are not available in the high-level description may provide a richer semantic description of the image, which are often quite powerful for content-based indexing. Semantic concepts capture visual low-level features like the green color and texture of the field and allow it to be described as *grass*. Similarly, the collection of all low-level features for the grass, athletes, and crowd can be described as *soccer*. Semantic concepts can represent specific objects like *cars* or *people* or general scenes like *sunset* or *forest*. Finally, as illustrated these conceptual levels of features exist at different points in the continuums of understanding for either humans or machines. Due to this understanding disparity, research in the field has become splintered into a few discrete schools of thought, summarized below from detailed discussions [45],[24].

- **Let the internet do it.** Given the abundance of Web users and their willingness to perform simple tasks, some propose that with a number of labeling games [119], all images can be sufficiently tagged by human participants with a small set of words, which thereby reduces the need for any machine-based processing. Even if only a fraction of the online population participates in these games, the availability of multiple high-level, human-based tags still offer an immediate set of textual features that can be indexed and searched with existing algorithms.

- **Learn the objects first.** Computer vision techniques have long focused on creating exact, physical representations of a particular object. For example, an image of a restaurant could be described in a machine by some of the features described in section 2.1.2 or by a composite of multiple objects like a cup, table, chair, plate, food, etc. Even with the computational resources available today, this detailed and highly granular analysis of scenes is not entirely practical because it requires a very large visual *ontology* (also called a dictionary or library) of objects and each of those object may have many unique low-level appearances (i.e. colors of cups or shapes of tables) which add complexity to learning a single machine-based model.

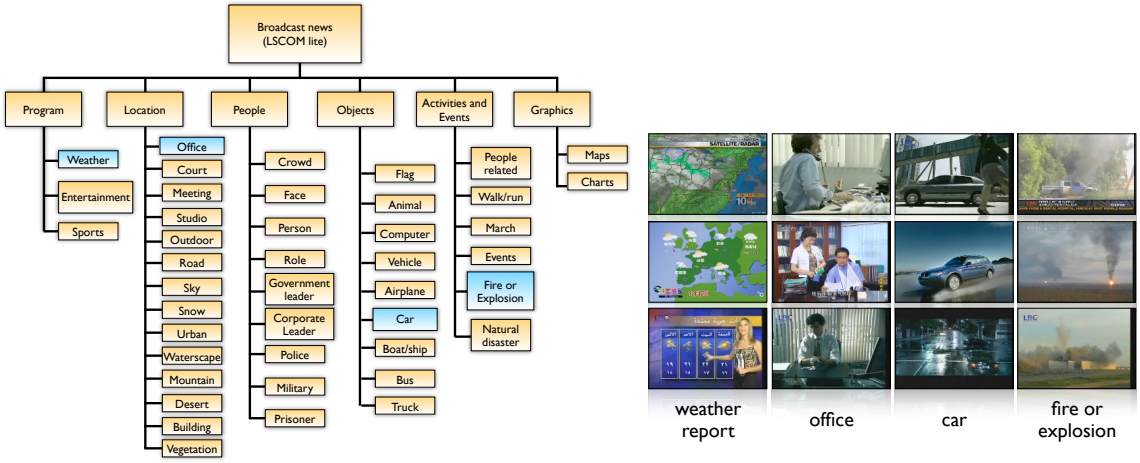
- **Map feature regions to words.** Conceding that learning a full library of objects is too complex, some approaches have sought to map a small set of low-level features, or regions of features, to parts of objects and then generate numerous visual descriptions

from these more elemental parts [67]. Similarly, one approach called ImageNet leverages the textual object ontology WordNet [31] to build a similarly structured visual ontology by crawling Web images that correspond to each object category [27]. Another approach uses a large number of images, a nearest-neighbor retrieval of regions with low-level features and a hybrid of probabilistic inferencing and affine warping processes to transfer region labels to new content [68]. Both of these approaches require a large number of crawled Web images, however the former task requires only noisy image-level annotation while the latter task requires a supervised annotation of regions within crawled images.

- **Learn a set of semantic-based concepts well.** This approach acknowledges the difficulty and nuances of all of the above tasks and instead suggest that a small set of mid-level semantic concepts that should be learned with a high degree of accuracy. Similar to the above approaches, large ontologies of 1000 [57] and 500 [105] visual concepts have been defined by a panel of experts and potential users. However, unlike the data-driven approaches for formulating an ontology above, this ontology is well-formed and realistic because it was explicitly defined by experts and users. The drawback of this approach is that from this one solution, two problems are created: utilizing low-level features to describe these semantics and a burden on the user who now must inspect the available ontology and choose those concepts that best match his or her search topic.

Considering the benefits of each practice, the work in this thesis falls into the last, hybridized school of thought. Although a user must now be informed about available concepts, this approach avoids potential pitfalls from other solutions by avoiding noisy labels from the internet, not requiring a multitude of models for specific objects, and applying semantics to scenes and objects instead of arbitrarily defined image regions. Multimedia search is a process that inherently involves a user and a search topic so it is intuitive for a user to choose from a set of semantic concepts that best answer the search topic. Exploring the definition of LSCOM (Large Scale Concept Ontology for Multimedia [57]), each concept is first assigned to one of six categories: program, location, people, object, events, and graphics; corresponding examples for each of these categories are “weather report”, “office”,

"face", "car", "explosion", and "charts". A subset of 39 concepts and corresponding image examples are shown in figure 2.3.



(a) (b)

Figure 2.3: Using the LSCOM ontology and TRECVID2005 dataset, (a) an organized chart of 39 semantic concepts and (b) diverse, multi-lingual image examples for four selected concepts.

When properly defined, semantic concepts describe a wide range of visual appearances, so creating a new concept is not a trivial process. Each concept must be defined and annotated by humans and then trained by machine to produce a model for classification, as described in the remainder of this section.

1. The ontology of semantic concepts must be defined and the name and textual definition of each visual concept should be created so that everyone using the ontology agrees upon a consistent meaning for its content. The LSCOM ontology is composed of 1000 semantic concepts [57]. This ontology was defined by a collection of experts in the computer vision, computational linguistic, library science communities and "user" community groups like government analysts with the requirement that each concept be useful (realistic for video search), feasible (capable of being detected), and observable (frequent in video datasets and observable by humans). Observability of a concept not only accounts for the raw number of instances in a dataset. It also takes into account whether a concept is observable by a human user by simply inspecting

the audio-visual content without needing additional background knowledge (e.g., the world series championship game vs. a baseball game).

2. With a pre-defined dataset, human labelers are asked to inspect every image in the dataset and annotate it with a relevant or non-relevant label for each concept in the ontology. A subset of the LSCOM ontology was used to exhaustively label a set of multi-lingual broadcast news and train semantic concept models defined as *Columbia374* [129].

3. Low-level features, like those described in section 2.1.2 are computed for the dataset at-hand. Any low-level feature can be used, but generally a combination of both global and local features produces the best results in empirical studies [54].

4. Machine learning algorithms that *classify* content like SVMs (support vector machines [22],[15]) or HMMs (hidden Markov models [91]) are learned in a *training* mode so that analytical models can be constructed to capture the distinct properties of images that were annotated as relevant and non-relevant. Using typical training procedures to partition the data into "training" and "testing" subsets, the performance of a model can be evaluated (deferred until section 2.3). Higher performance scores indicate that a particular model will be more accurate when used to classify unseen images.

5. To classify (or evaluate) any machine-learned model on new data, repeat this process starting from step 3, using classifiers in their respective *testing* mode. The classification score for every image-concept pair indicates a confidence score (or relevance assessment), $p(x_n; t)$ for one image, n, with content relevant to one concept, t. Ideally, this relevance is probabilistic and has a score between zero and one, but SVMs require a post-classification normalization to abide by this range. In this work, a sigmoid normalization is utilized, where a sample's raw score $s_{n,t}$ for concept t is normalized with the equation $p(x_n; t) = \frac{1}{1+exp(-s_{n,t})}$.

Once an ontology is defined and annotated it can be used in a largely automatic sense. An advantage of mid-level concept classifier scores over low-level features is that each image-concept pair has only one piece of information to store: a confidence score. Thus, whereas

additional computations are required when searching among low-level features, concept scores can be absolutely compared to each other. More specifically, if a user was interested in finding images that were related to the concept "car", a system only needs to sort the confidence scores for "car" and return those results.

2.2 Content-Based Search

Multimedia search evolved as documents grew from words to include images and sounds. Multimedia search systems are rooted in the information retrieval (IR) community, so principles like the execution of a search and performance metrics like precision and recall are common between the two. A system executes a search, or the process of finding relevant multimedia documents, given a specific query, q. A query defines the properties of a search, which may come in the form of keywords, images, or semantic concepts. To this day, neither systems nor experts can construct perfect queriers for arbitrary search topics, but methods for both automatically and interactively doing so are explored in the next chapter in section 3.1. This section introduces common evaluation metrics for multimedia search and demonstrates how content- and concept-based indexing is used to execute a search.

2.2.1 Query by Content

When querying with a content example, the system attempts to identify documents within a dataset that most closely match the specified query. For text-only documents, like Web pages, content is simply the words in that document. One the most commonly used algorithms to measure textual similarity of a query to other documents is *TF-IDF*, which computes the ratio of term frequency (the frequency of the query word in a document) to document frequency (the number of documents that the word appears in). For experiments in this thesis, a more robust definition that normalizes scores by document length, called pivoted TF-IDF, is used for all searches with textual queries [98]. *CBIR*, or content-based image retrieval most commonly uses visual low-level features to formulate a query. With visual low-level features, a database, X, of N images can be indexed by vectors of dimension D, $X = \{x_1, x_2, ..., x_n\} \subset \mathbb{R}^D$. With this representation, the distance between a query

image and any image in X can be computed. Typically, the scores derived in this step of the CBIR process are quite difficult to understand by humans, so the appropriate distance metric must be chosen for any given dataset. Common distance metrics used in this work are the Euclidean or $L2$ (equation 2.1) and cosine (equation 2.2) metrics.

$$dist_{Euclidean}(x, y) = \sqrt{\sum |x_d - y_d|^2} \tag{2.1}$$

$$dist_{cosine}(x, y) = 1 - \left[\frac{\sum x_d \; y_d}{\sqrt{\sum (x_d)^2} \sqrt{\sum (y_d)^2}} \right] \tag{2.2}$$

During a search, the distances between a query image and all images in a database are computed and returned in increasing order, giving images more distant from the query a higher enumerated position. However, to align these results with the notion of relevance, it is often desirable to transform a distance into a similarity score. The most naïve technique for this is to simply flip the meaning of the score range: $similarity = 1 - distance$. Distances not in the range of zero to one, like Euclidean distances, may first be normalized by the maximum distance to a query image. While these distance metrics may be simple, to accommodate large datasets and minimize computation time, search engines often directly index the pair-wise similarity scores between images of a database instead of the low-level features themselves.

2.2.2 Query by Semantic Concept

Searching with semantic concepts is one way to bridge the semantic gap between visual low-level features and high-level human understandings of an image. As discussed in section 2.1.3, a pre-defined concept ontology allows a multitude of visual concepts to be defined and used to index a dataset. In this work, a subset of the LSCOM ontology is selected and the TRECVID2005 development dataset (multi-lingual broadcast news) was exhaustively annotated to train a set of models called the *Columbia374* [129]. With a rich set of pre-trained concept models, new multimedia clips are segmented into subshots, classified by each concept model, and indexed by those concept classification scores. When a concept-based query is evaluated, the system retrieves the scores for all subshots and the appropriate concept and creates a list sorted in decreasing order such that subshots with a higher classifier score are correctly assigned a lower rank. For this reason, a concept-based query

is more intuitive than a content-based query because a single classifier score is directly presented as a relevance score instead of computing a secondary similarity score based on low-level features; this property alone helps to close the semantic gap inherent in multimedia indexing.

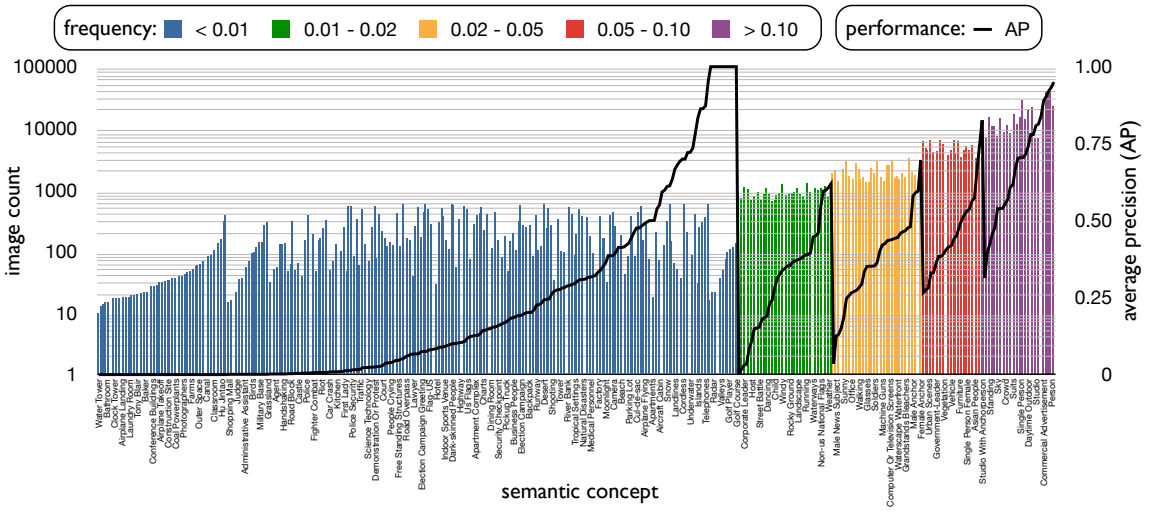


Figure 2.4: Count of training images (from a set of 65,256) and average precision of the classifiers over all concepts in the *Columbia374*; concepts are ordered by increasing performance and grouped by training image frequency. Most concepts have over one hundred unique training examples, but classifier performance varies dramatically with the difficulty of each concept.

Another important benefit of concept-based queries is that concept classifier scores are produced from machine-learned models that have analyzed tens of thousands of image examples. This extra layer of machine-learning allows the model to be more robust and produce an accurate classifier score (as measured by average precision, introduced in section 2.3) for two different images that contain the same relevant concept, as shown with the image examples in figure 2.3(b). Considering these concepts as potential query inputs, it is not surprising that performance of a content-based query generally falls short when compared to that of a well-trained concept classifier. Looking more closely for a cause, figure 2.4 demonstrates that most concepts classifiers in the Columbia374 were trained with over one hundred unique image examples. However, as indicated by the first (blue)

concept grouping, less than 1% of the 61901 training images from TRECVID2005 dataset had content that was relevant to the majority of concepts. Also observable in this figure is that even with thousands of examples, the performance of different concepts can vary dramatically according to the underlying complexity of that concept. Complexity in a visual concept usually stems from the concept's object being too small (i.e. a "phone" in an office setting), the concept having appearances that are too diverse (i.e. the different appearances of a "flower"), or the variation in imaging environment and condition (i.e. the variety of backgrounds with a "corporate leader", lighting conditions, and partial visual occlusion).

Just as a single image can contain multiple relevant concepts, queries can be formulated with multiple concepts. These formulations can be with simple logical constructs to facilitate search breadth both wide (i.e. results with one or more relevant concepts from a set) and narrow (i.e. results with all relevant concepts from a set). The ability to quickly construct classifier conglomerates creates a rich tool for search systems that can be leveraged to search and filter content in a number of simple ways [136].

2.3 Evaluating Multimedia Search

Once a query is formulated and a search is executed, its performance (i.e. recall and precision) can be scored after ordering results by decreasing relevance score. Analyzing the set of returned search results, R, and the entire database, X, recall indicates the fraction of relevant results returned compared to all relevant results in the database and precision indicates the fraction of results relevant in those returned. The function $Rel(q, X)$ returns the count of results in X that are relevant to a particular query q.

$$Recall = \frac{Rel(q, R)}{Rel(q, X)} \tag{2.3}$$

$$Precision = \frac{Rel(q, R)}{|R|} \tag{2.4}$$

Other popular metrics in multimedia retrieval systems include the precision at a certain depth or $P@D$ where D is the pre-specified depth, and average precision (AP), which averages precision measured at every hit position up to a certain limit. For search tasks, both of these metrics are better suited than pure precision and recall because they reward

lists that have more relevant results at the top (i.e. low depth).

$$Precision(D) = \frac{Rel(q, \{r_1, ..., r_D\})}{D} \quad (2.5)$$

$$AP(N) = \frac{\sum^N Precision(n) Rel(q, \{R_n\})}{Rel(q, X)} \quad (2.6)$$

One surprise to readers unfamiliar with the metric is how low AP scores can be. Whereas a high precision score would be in the range of 0.8-1.0, a high average precision (at a depth of 1000) might reside in the range 0.2-0.4. It is important to realize that this difference is caused by the averaging of precision scores, and in terms of research progress, AP gains within of only 5-10 points are generally considered significant. In some information retrieval works, AP has also been interpreted as the approximate area under the precision-recall curve.

Objectively evaluating a multimedia search system is non-trivial. Often, the task is complicated when collecting a dataset with distributions matching real-world scenarios, minimizing subjectivity in query topics, and requiring large amounts of human inspection time to evaluate each search result for each query topic. TRECVID, founded in 2003, is an annual independent evaluation of multimedia tasks ranging from shot detection, concept detection, and video search that pragmatically answers each of these concerns by the National Institute of Standards and Technology (NIST) [101]. Since its foundation, TRECVID has driven research for many techniques by offering objective evaluations over fixed datasets and identical tasks. TRECVID datasets from 2003-2006 were each composed of 100-150 hours of international news broadcasts, providing high-quality video and speech transcripts with well-defined editing styles. TRECVID datasets from 2007-2009 were each composed of 150-200 hours of Dutch documentary and entertainment broadcasts providing lower-quality video and less uniform editing styles.

Each year, NIST objectively constructs 24-48 TRECVID search topics with content examples either defined by temporal references into the dataset or static images from the Web. For each year, the search topics are best answered by different aspects of multimedia search allowing different systems to truly discover their utility with different query types. For example the topic “Find shots of Tony Blair” (TV05) may utilize text and near-duplicate queries, “Find shots of scenes with snow” (TV06) may utilize a concept query, and “Find shots of one or more people, each in the process of sitting down in a chair” (TV08) may

utilize a hybrid of concept and motion-based similarity queries. Search topics are released to a pre-registered community and the top 1000 search results for each search topic are collected roughly one month afterwards. Each community member can try to answer the search topics with systems that are fully automated (no human interaction), manually configured (a human specifies the query), or fully interactive (a human is fully engaged). After collection, results from all members are pooled for each search topic and a subset of those results is inspected by independent evaluators at NIST. Finally, NIST returns relevance assessments and scores each member's search submissions with the AP metric. Result pooling for relevance inspection has one major, and potentially dangerous assumption: only pooled results are inspected so all relevant results for a query must have been returned by at leas one community member. Much of the work in this thesis and all of the semantic concept models are evaluated with benchmark data and ground truth annotations provided by NIST and the TRECVID community.

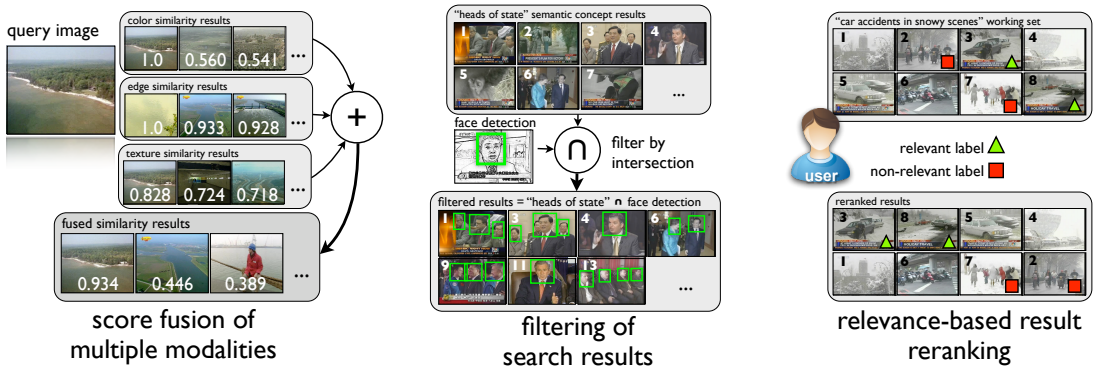
2.4 Multimodal Search

Given the basic building blocks for content- and semantic concept-based queries, systems can answer search topics with targeted multimodal queries that leverage the relevance from many sources. Searching with one or more modalities can lead to queries that are quite complex, but the goal is always to find the most relevant results from a dataset. Illustrated in 2.5, this section discusses the following common strategies for utilizing sets of scored results from multiple modalities: fusion, filtering, and reranking.

Fusion The most straight-forward technique for combining the results from different modalities is result fusion. As previously described, queries using low-level features express similarity to a query, R_{sim}, and queries using semantic concepts express classifier probability for a concept, $R_{concept}$,.

$$R_{sim}(X, q) = sim(q, x_n) \tag{2.7}$$

$$R_{concept}(X, t) = p(x_n; t) \tag{2.8}$$



(a) (b) (c)

Figure 2.5: Strategies for combining search results or user input: (a) multimodal fusion, (b) result filtering, and (c) relevance-based reranking.

Result fusion combines the image relevance of multiple modalities, K, into a single list without additional knowledge about a modality source. This fusion can be accomplished either by directly combining the relevance score or rank (i.e. position in the list) for each member of X. Additionally, one of many linear and non-linear methods (like *min*, *max*, *sum*, etc.) for combining these lists has also been analyzed [9]. Subsequent research found that regardless of the mathematical operation utilized, treating each result list equally often lead to poorer performance. Parallel works found that a form of linear weighting for each result list, here optimized by analyzing a set of training queries, was often quite powerful [7]. Preference, either pre-computed from training data or assigned by a user, can be given to one modality over another by applying different modality weights, ω_k, during fusion. This may advantageous if the performance of a concept query were presumed to be better than a keyword query. Equation 2.9 demonstrates a typical fusion strategy including different modality weightings.

$$Rel_{fused}(X) = \sum^{N}\sum^{K} \omega_k R(x_{n,k}) \tag{2.9}$$

The choice of proper fusion weight is a challenging topic of its own (often referred to as "feature selection") and not within the scope of this thesis. Finally, in this work fusion is computed with the normalized relevance scores of each result list $R(X)$ with the equation

below.

$$R_{norm}(X) = \frac{R(X) - min(R(X))}{max(R(X)) - min(R(X))} \tag{2.10}$$

Filtering Result filtering allows simple logical rules to be constructed and applied with different search modalities. Filtering is generally useful when a user wants results with high precision and with a possible sacrifice of recall. The most common logical rules are union, exclusion, intersection, and difference. After filtering, the final relevance list for this example should include only the faces of heads of state, which more closely agrees with the example search topic. For example, if the search topic is "Find shots of politicians speaking", a sensible search strategy would be to first search with the semantic concept *heads of state* and then filter (by intersection) with the relevance scores of a face detection algorithm. Similarly, for the search topic "Find shots of basketball players on the court", a sensible strategy might use a union of the concept *grand-stands* and *athlete* but exclude results from the concepts *golf* and *outdoors*.

Reranking Reranking analyzes the results R from one search (called the *working set*) and rescores the images according to user relevance labels or secondary low-level features and concept classifier scores. Once new relevance scores are computed, they are combined with the initial scores for the working set with a pre-determined weight, α, according to the equation below.

$$s_{reranked} = (1 - \alpha)s_{initial} + \alpha s_{rescored} \tag{2.11}$$

The variable α facilitates a trade-off between the "stable" initial relevance scores and the relevance scores from the "dynamic" structure produced from other low-level features or concept scores. Typically, the goal of result reranking is to greatly improve the precision of a working set with only a few user-provided relevance labels where the recall for a set of results is fixed as no additional results are retrieved from the database at large. Result reranking can be executed both automatically or in user-solicited *relevance feedback* environments, as was first implemented in text-based information retrieval systems [130]. This topic is of particular interest in this thesis and its discussion is continued in the next chapter.

2.5 Summary

In this chapter, an overview of multimedia indexing and search systems was provided. Much like human understanding, a machine's representation of multimedia content can range from low-level features (such as words or colors) to mid-level semantic concepts (such as general objects, events, and scenes) to high-level descriptions (such as named entities). As by-products of machine-based multimedia indexing, gaps in sensory and semantic perception permeate each of these granularities, so a novel mid-level representation, referred to as semantic concepts is introduced. With these various representations, a well-defined and automated pipeline for segmentation, feature extraction, and computation of concept confidence scores for new multimedia documents is reviewed. To evaluate performance of an indexing system, common metrics from the information retrieval and machine-learning communities like precision, recall, and average precision are were introduced. Finally, methods to search an indexed multimedia dataset using specific content queries were discussed along with a few strategies for combining the results of multiple queries. With these foundations, subsequent chapters explore common difficulties for interactive search systems and how to gracefully assist users during the entire process.

Chapter 3

Review: Interactive Video Search and Relevance Feedback

In the previous chapter, an introduction to common multimedia search tools and their usage in a system was discussed. Compared to fully automated search using visual low-level features or semantic concepts selected by a system's "best-guess", the results are often dramatically improved when a human is involved in the process. Contrary to automatic multimedia search, a user is allowed to not only construct the query but also refine it during the result inspection stage, thereby guiding the system to the search target and correcting missteps along the way. Within this thesis, human actions are required for three parts of an interactive search: query formulation, result browsing, and relevance assessment. This chapter reviews each of these activities and advances from both automatic and interactive search environments. Section 3.1 describes how a user can create a fully formed query from a search topic alone. Section 3.2 reviews methods utilized during the inspection of results and query refinement. During result inspection, systems often employ methods for relevance feedback, discussed in section 3.3, which help the machine better approximate the user's search target. Finally, open problems regarding each of these topics are reviewed in 3.4.

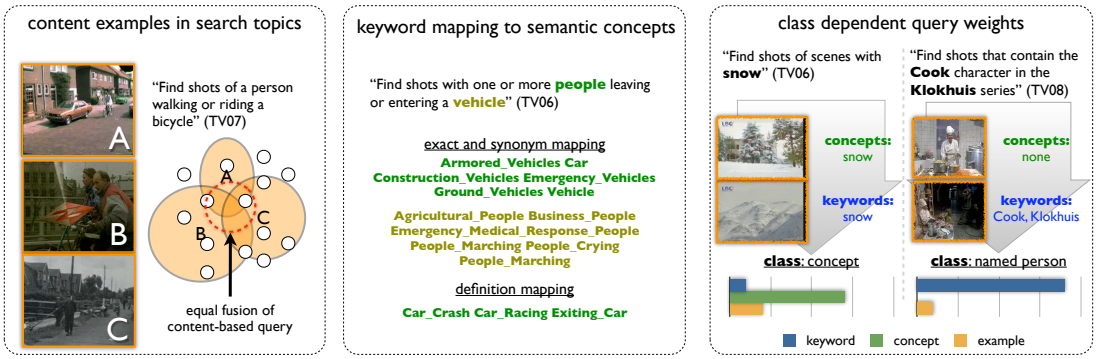
3.1 Query Formulation

Query formulation can be equated to the planning stage of any problem solving technique. A search topic defines a specific information request for the user, whereas the query, constructed by a user or automatically, tells the system what initial steps to take to best answer that search topic. Query formulation is non-trival because it requires the selection of query modalities like low-level features, semantic concepts, or others computed during content indexing (as introduced in section 2.1) and the proper weighting of each modality as it relates to the entire query. This section first reviews various automated query formulation techniques and then explores the different methods available to users to interactively define their query.

3.1.1 Automated Query Formulation

In automated query formulation, the system is directly given the text or image criterion that define a search topic. With this information, the system begins to lexically and statistically *expand* and *map* the criterion to query parameters. Expansion is the process of deriving multiple keywords or images from one original source through correlation information. Mapping is the process that helps to resolve the disparity between two sets of entities. For example, between the sets of colors {*red, blue, green*} and {*orange*}, the best mapping might be *red* $\Leftrightarrow$ *orange*. Using previous TRECVID query topics (introduced in section 2.3), figure 3.1 illustrates the needs and a few possible solutions for automatic query formulation.

Image examples provided with a search topic can be utilized directly with content-based query methods. In this fashion, a reasonable query formulation might consist of an equal weighting of visual modalities for the given image. For example, if the search topic is "Find shots of a person walking or riding a bicycle" (TV07) with one or more image examples, the system can use low-level color and local image features for search. In a fully automatic scenario, the system must combine evidence from these images to best compute search relevance. One of the simplest combination methods is the fusion of these different modalities with equal weights (section 2.4). In another work [80], multiple overlapping classifiers were trained using several search topic examples. For example, if five



(a) (b) (c)

Figure 3.1: Example search topics and potential solutions via automatic query formulation: (a) fusion of content examples, (b) expansion and mapping of search topic, and (c) query-class dependent weighting.

topic examples were provided, instead of training a single classifier with all five samples, the authors would train three or four models from randomly selected subsets of these examples. Afterwards, each model is evaluated on the result dataset and there scores are averaged. An interesting aspect of this work is the use of topic examples as relevant results and randomly selected members of the dataset as non-relevant results. Although odd at first, non-relevance to a query is often ill-defined and seldom provided in information retrieval tasks. In the example above, a list of the semantic objects that are *not* bicycles would be infinitely long. Also, most realistic datasets have a low prior frequency (or number of relevant results) for any one specific search target. Looking at the prior frequency of the Columbia374 semantic concepts in figure 2.4, only 29 of these concepts were present in 10% or more of the TRECVID2005 dataset. For these two reasons, random selection of dataset members is often a reasonable approach for collecting non-relevant examples for machine learning tasks.

Textual keywords provided for a search topic are also equally important and can be utilized with keyword expansion and mapping. Although this thesis downplays the fitness of keyword-based queries for multimedia search topics, several lexical tools can help to expand a simple description into a rich set of semantic concepts. One of the most commonly utilized tools is WordNet, a lexicon of English 155,287 words that has a rich ontology

from hyper/hyponyms (the *is-a* relationship) and synonyms (similar meanings) and useful full-text descriptions. Combining the synonyms of WordNet with keyword expansions from other data sources (like ontological definitions [57], simulated text searches [43], online news articles [20], or photo sharing sites [126]), a small number of keywords can be expanded and mapped into a large number of semantic concepts. For the search topic "Find shots with one or more people leaving or entering a vehicle" it is advantageous to expand and map the words "people" and "vehicle" to many different semantic concepts. After mapping, a list of concepts can be culled through mutual information or other statistical analysis techniques to validate that the mapped concepts coarsely agree. One significant drawback of keyword expansion and mapping techniques is that a large amount of tuning and customization must be performed to avoid adding too much noise in the form of extraneous relationships.

Acknowledging that mulitmedia search topics can be defined with both textual keywords and content examples, many works have been devoted to unifying the automated processes for both. This topic, often referred to as *query-class mapping* analyzes all components of a search topic and assigns (often linear) fusion weights to the different parts of the resulting query. First, via analysis of several training query topics, query classes are automatically derived or manually chosen using a latent semantic analysis [128] or joint query semantics and performance clustering [60] to produce a handful of query classes like "named person", "sports", "concept", "named person+concept", "named object", "scene", or "general". During this class formulation, weights for different query parameters (i.e. text, concept, and example) are pre-computed. Later, when a new search topic is received, the topic is analyzed for keywords and semantic concept similarity and the most similar query classes is chosen and the fully-formed query is evaluated with the class's pre-computed weights. Figure 3.1(c) demonstrates how two topics "Find shots of scenes with snow" (TV06) and "Find shots that contain the Cook character in the Klokhuis series" (TV08) are mapped to unique query classes. The automated learning and application of query classes offer promise to the challenging class of problems involving feature selection (which modality to use). Unfortunately, most query class approaches have been shown to be highly sensitive to training queries and the number of classes learned. Observing this limitation, recent work used keyword mapping after disambiguation, distribution normalization for concept

classifiers, and data mining for content examples avoiding explicit mapping to a single class or a single set of fixed weights [79].

Automated query formulation uses lexical and statistical tools to compute the best weights for a set of query parameters. These techniques leverage different modalities and discover data relationships that may not be obvious to the casual searcher. However, with each fully automated technique, authors often cite the lack of an approach to all queries in general and need to retrain their automated method for each new set. In most cases, an interactive search user can leverage the fully automated "starting point" and further reduce ambiguity problems from the search topic.

3.1.2 Interactive Query Definition

If two strangers approached each other and one asked for directions to the nearest bank, would the other respond by drawing a map, pointing in a specific direction, or vocally enumerating a set of turns? Clearly, the modality used to answer this request would depend on available resources and familiarity with the request itself. The same statement is true for interactive video search. As defined in the last chapter, a diverse set of features and modalities are available to formulate a multimedia query; how they are used to do so is truly up to the user.

Systems to perform content-based image retrieval have been devised since images content was first made available to machines. One pioneer of CBIR techniques was the QBIC system [32], which used user-specified color histograms. In this system, the query is formulated only using a color spectrum and the relative importance of each color in that spectrum. This system largely succeeded for 2-dimensional images and drawings, but shortcomings were exposed with more complex images because the system could coarsely match percentages of color for the whole image, but it did not know how those colors should be distributed in the result images (i.e. an object, a region, or the entire scene). Following this work, effort was shifted to exactly solve this problem by letting the user sketch his or her query. Using basic shape contours [26] and a hybrid of color, shape, and texture operators [103], accuracy in CBIR systems grew because the system knew more about the desired search target. However, perhaps due to the semantic gap (introduced in section 2.1.3), results in

the system would only sporadically be correct. At this time the system had no proxy to guarantee that the results had any coherent semantics. For example, a query for underwater images has a user-sketch with large regions of blue and green, but the search results of this query contain underwater images as well as outdoor scenes with majestic blue skies and lush waterfalls in a rainforest. The efficient use of image examples and performance of CBIR systems continues to improve with the addition of more discriminative features. Capitalizing on this fact and technological evolution placing digital cameras on modern cell phones, near-duplicate matching services to facilitate product purchases [33] and find directions [65] have been created.

Recognizing the difficultly for most users in describing the search target by its parts or directly acquiring actual examples of the target, active research in the multimedia search community has largely shifted to representing images and objects within images with words. Debate over the best learning and labeling methods remain unanswered (section 2.1.3), but with text-based input, either by keyboard or speech, currently the preferred query input strategy, most search systems now utilize tools from the text community to lexically map user keywords to names of objects and semantic concepts. It is no coincidence that the automated methods reviewed in the last section are directly complementary to this form of textual query specification. In fact, text to concept mapping and other data-mining techniques can be quite beneficial to users when applied responsibly. Alternative methods to select words or semantic concepts for a query have also been presented, like alphabetical listings or multiple checkboxes, but empirically users are overloaded by the quantity of available concepts so these methods are not widely adopted.

3.2 Result Browsing

Both automated and interactive forms of query formulation were discussed in the last section. Once a query is formulated and evaluated, the main topics of concern are intuitive result presentation and methods to assist the user to better organize and traverse a large set of results.

3.2.1 Intuitive Result Display

Following the standard practice set forth by current Web search engines, most results for interactive multimedia systems are presented in simple list or grid forms. In these systems, a user understands that the most relevant result is placed in the top-left corner of a result page and relevance decreases row-wise scanning right and then down in the page. While these displays are both tolerable and now expected, alternative forms of display have also been proposed. A popular technique to more intuitively display results is to first cluster the results with low-level features and then alter their spatial position according to cluster membership. A result's new position can either be aligned to a visual grid [64] or free-form [69], [82]. In a subjectively evaluated work [69], results in the same cluster were spatially grouped during display to the user, which empirically assisted users in finding relevant images more quickly and with higher precision. This work also presented a behavioral study comparing a system that visually cropped regions of the image (according to a user-interest operator) and a system that visually enlarged the currently inspected image result (as determined by mouse interaction). In this work, users only liked the cropping technique but cited errors from the automated operator. Instead of disrupting the visual grid that users were accustomed to, another work reorganized the visual with a large set of semantic concepts to construct meaningful semantic clusters called *Visual Islands* via dimensionality reduction and entropy measurement [135]. Again, experiments confirmed that clustering, now among visual semantics instead of low level features, was helpful for speed and performance and demonstrated increased ability to remember the semantics of inspected results over non-organized results. Each of these works focus on changing the result display to more intuitive to the user, and some have the ability to dynamically update their displays according to user input. However, few of the above systems allow the user to quickly traverse the entire dataset. Instead, most systems focus on a smaller pre-computed *working* result set, so it is difficult for a user to ascertain quality of the current query with respect to the entire dataset.

3.2.2 Organization Interaction

Complementary to intuitive result display is the ability to intelligently traverse a dataset with respect to those results. Most systems have resorted to using tree or folder-based paradigms that presume some existing ontology, perhaps due to the nature of the task and user familiarity with the construct. Now a part of the MPEG-7 standard, early work proposed that a number of different tags (or textual categories) be available and embedded in user media so that any search system could quickly recognize these tags [11]. Similarly, other early works adopted this tree-based approach [61], presented shared category and calendar viewpoints [1], and allowed users to dynamically define tags based directly on sets of results [110]. Beyond these data-driven organizational methods, modern works frequently tie a user's organizational efforts directly to his or her interaction. Techniques have experimented with alternative multi-user display configurations [89] and keyboard-free interactive surfaces like table-top displays [66] or distributed smart phones [99]. All of these approaches reduce the dependence of a single user on the relevance computations and display of a single machine. Instead, one or more users assist each other and can move content to a different region of the system to be re-inspected at a later time or closely examined by an expert. However, as the lines between multimedia search and interaction blur, particularly with the introduction of multiple users, it is reasonable to question the effectiveness of multiple human searchers. Currently, dramatic performance increases over a well-designed, state-of-the-art automatic or semi-automatic system have not been fully demonstrated.

If the goal of an image retrieval task is shifted to "experience augmentation" instead of answering a specific search topic, unique applications can be created in mobile environments. Constantly searching the online photo site flickr [124], results were organized by general topics (i.e. "topic of the day"), personal experiences, established social connections to other users (i.e. searching for others' experiences), and by geo-spatial information that changes depending on the location of the user. In a similar approach authors of the system in [65] also organize results by location but created an "augmented reality" display with the results to facilitate exploration of highly-localized content (i.e. the inside of a specific house or the different appearances of a single tree over multiple seasons). This form and application of multimedia retrieval is still in its infancy, but the growth of digital social

communities and asynchronous mobile technologies already guarantees its future.

3.3 Relevance Feedback

Advanced methods for query expression and search may be available but often a system can never perfectly match a user's search target. Relevance feedback is an iterative query refinement technique that the system can use to better approximate the user's search target with relevance (or user preference) assessments. The two main issues regarding relevance feedback are: how is the feedback acquired and what does the system do with the feedback once it is available. The remainder of this section discusses common answers to these questions and provides an introduction to relevant approaches.

3.3.1 Acquiring User Feedback

User feedback describes the relevance of a result to the user's search target and is commonly acquired explicitly or implicitly. In this thesis, explicit feedback is a single user label for a search result, either relevant or non-relevant. Implicit feedback can also use a binary relevant or non-relevant label, but is also derived from the order of search results from a user's query.

3.3.1.1 Explicit, Solicited Feedback

Relevance feedback was first utilized as a method to assuage the burden of result inspection. In an early relevance feedback work, a two-layer set of algorithms was applied after a user was asked to rank a small set of results [94]. The first layer evaluated different feature modalities and distance metrics to select the parameters for each that produced the highest rank agreement with the user's annotation. The second layer would then normalize individual feature values according to statistical distributions among reranked results. While this system was fast enough for real-time evaluations, it required the somewhat awkward process of reordering a subset of results, which may be difficult or arbitrary if all members of this subset were non-relevant. Recognizing this shortcoming, a new approach called *active learning* was introduced, which utilized relevance feedback to solicit relevance labels

for the most informative result. In an active learning work [112], SVM (support vector machine) models were trained with low-level features and relevant (positive) or non-relevant (negative) labels. The SVM model was then used to reclassify all unseen samples. Those samples with scores closest to zero (i.e. lowest confidence) were "actively" selected the next stage of relevance solicitation from the user. Utilizing a small number of iterations and solicitations, the machine effectively decreased its confusion about the search target and increased relevance of results, as determined by classifier scores. In a work that efficiently loaded results and features in a real-time system, multiple rounds of relevance feedback and active learning were interchangeably executed (even at the user's request) to quickly explore a large portion of the target dataset and achieve interactive high AP scores [81].

Active learning has also been successful when used to select samples for large-scale label annotation tasks, such as those required for annotation of the new TRECVID datasets. While solving this realistic problem in these datasets, one work analyzed different techniques to solicit annotations [4]. Using the 21k keyframes from TRECVID2007 dataset of documentary and educational video, 36 semantic concepts selected by NIST were fully annotated facilitating a few interesting observations summarized below.

- Between sampling techniques using uncertainty-based sampling (optimizing precision) and relevance-based sampling (optimizing recall) shown in figure 3.2, easy search targets (semantic concepts in these experiments) demonstrated relevance sampling is best with less than 15% of the database annotated and uncertainty best with more than 15%; moderate to difficult search targets showed no preference between sampling methods. This sampling ambivalence for moderately difficult concepts indicates that either the learned models were too poor (i.e. the utilized features did not sufficiently describe the content) or the samples were largely inseparable (i.e. the original feature space contained samples that were very similar but belonged to both the relevant and non-relevant classes).

- Prior models trained on slightly related data can help to prevent "cold-start" problems when annotating new data. This finding validates that when re-annotating existing concepts for a new domain, old models are still quite useful.

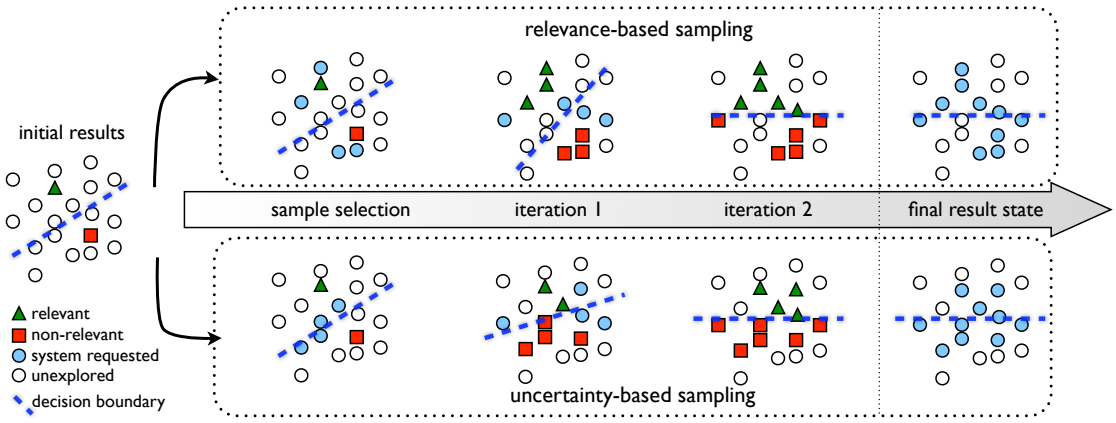


Figure 3.2: Two active learning strategies and their coverage of a search result set. Even with careful sample selection, both techniques require many labels and can still miss regions of the result space with relevant samples.

- The inclusion of multiple features, even if poor in quality (i.e. text features or temporal neighbors), can help active learning.
- With active learning, near-peak classifier performance could be achieved with the annotation of as few as 15% or as much as 50% of the total keyframes in a dataset. This performance guarantee comes from the fact that sample selection with active learning has generally found over 80% of the relevant samples in a dataset with less than 40% of the data annotated.

Finally, in a state-of-the-art application of the lessons from this annotation effort, a system was trained to dynamically propose relevance-, uncertainty-, or temporally-based sampling techniques during interactive search sessions on TRECVID2006 and boosted the mean average precision (MAP) score from a 0.07 baseline to an impressive 0.32 peak [75].

While relevance feedback has proven to be quite powerful for both search and annotation tasks, it usually requires a large amount of training labels. Active learning methods can reduce this label requirement to 15% of an entire dataset with no significant performance loss [4], but with datasets that contain tens or hundreds of thousands of results, this inspection requirement could be quite taxing for a real-time interactive session.

3.3.1.2 Implicit Feedback from User Actions and Result Ranking

Implicit relevance feedback uses relative differences in result order and scores to ascertain relevance to a user's query. Users construct a query to align to his or her search target, so upon evaluation of this query those results with higher scores (i.e. concept confidence, image similarity, etc.) should also be more relevant to the search target. In an ideal setting, these results are perfectly scored and a pseudo-label of relevance or non-relevance can be inferred for each result depending on its rank after sorting. This technique is called pseudo-relevance feedback and can be quite powerful, depending of course on the quality of the search system and initial query. Due to this inferencing process, implicit relevance feedback may be weaker and sometimes incorrect. However, unlike explicit relevance feedback, pseudo-labels can be assigned to a far greater number of results (the entire result set) without user-consultation. In interactive search settings, additional methods for pseudo-relevance labels are available based on user actions like tabulating the time that a user inspects certain results, counting the number of inspected results with a query property, or the inspected locations of a results screen [41], [127].

3.3.2 Reranking with Relevance Feedback

Once explicit or implicit feedback is made available, relevance feedback systems must then decide how to utilize this additional information. This section introduces three related, but distinct technical areas that use relevance feedback to construct new models for reranking individual results, perform feature selection, or propagate these relevance labels to an entire dataset through result similarity. In all of the above cases, the goal is to reduce ambiguity for the search system and provide results most relevant to the user's search target.

Continuing this discussion, a more formal terminology for datasets and their representation is now provided. A single sample from a dataset X of size N is represented by its d dimensional feature vector x_i. Relevance labels for a single query over the entire dataset are Y and those for a specific sample are y_i. While each method in this section can produce scores for multiple relevance classes (i.e. classifying an image as inside vs. outside or dark vs. bright), the techniques in this section assume a single binary class with the labels $y_i = \{+1, -1\}$ representing relevant and non-relevant classes. Classifiers are binary

functions $f(x)$ that produce a continuous score depending on classifier confidence. Often, the sign of this score indicates membership in the appropriate class from above and its magnitude indicates class confidence. Finally, in every query, all samples in a dataset belong to either the labeled set X_l or unlabeled set X_u depending on the presence of a user-provided label.

3.3.2.1 Model-based Reranking

Model-based reranking is a subset of techniques that learn new classifier models utilizing provided relevance labels. Currently, the most prolific machine-learning approach is a support vector machine (SVM) [22], which produces a discriminative model that has the following definitions. A linear decision function $f(x) = w^T x + b$ creates a learned constant offset b and a hyperplane, or decision boundary, w that optimally separates the positive and negative training samples. In SVM's, sample points are easily compared in non-linear forms with the addition of a kernel mapping ϕ, to project a sample's features into a high-dimensional space $\phi(x)$. With no loss of generality, the new decision function becomes $f(x) = w^T \phi(x) + b$. In both cases, the hyperplane itself is determined by computing the largest margin of separation between different classes, as determined by the following equation.

$$\begin{aligned} &\min_{w,b,\epsilon} \frac{1}{2}||w||_2^2 + C \sum_{i=1}^{N_l} \epsilon_i \\ &s.t. \;\; y_i w^T \phi(x_i) + b \geq 1 - \epsilon_i, \;\; \epsilon_i \geq 0, \;\; \forall (x_i, y_i) \in X_l \end{aligned} \tag{3.1}$$

Here, ϵ_n is the penalizing variable added to each data vector x_i, and C determines how much classification error an SVM can tolerate. As proof of their utility, SVM's are used in this thesis to produce semantic concept confidence scores, as introduced in sections 2.1.3 and 3.3.1.1. One problem plaguing traditional SVM learning is that labeled sample set X_l must usually be quite large for the model to perform well, as demonstrated by the poor scores of many low-frequency concepts visualized in figure 2.4. Attempting to overcome this limitation, semi-supervised SVM's seek to leverage the labels from a small set of results by manipulating the distances between labeled samples according to their neighbors in the unlabeled space [12], [38]. In these works, the objective function is modified to select the

smallest penalizing error $(\hat{\epsilon}_j, \hat{\eta}_j)$ for unlabeled samples after assigning each sample to one of the two classes.

$$\begin{aligned} \min_{w,b,\epsilon,\hat{\epsilon},\hat{\eta}} \ & \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{N_l}\epsilon_i + \hat{C}\sum_{j=1}^{N_u} min(\hat{\epsilon}_j, \hat{\eta}_j) \\ s.t. \quad & y_i w^T\phi(x_i) + b \geq 1 - \epsilon_i,\ \epsilon_i \geq 0,\ \forall (x_i, y_i) \in X_l \\ & \hat{y}_i w^T\phi(\hat{x}_j) + b \geq 1 - \hat{\epsilon}_j,\ \hat{\epsilon}_j \geq 0,\ \forall (\hat{x}_j, \hat{y}_j) \in X_u \\ & -(\hat{y}_j w^T\phi(\hat{x}_j) + b) \geq 1 - \hat{\eta}_j,\ \hat{\eta}_j \geq 0,\ \forall (\hat{x}_j, \hat{y}_j) \in X_u \end{aligned} \tag{3.2}$$

Compared to SVM's, this method maximizes margin distance for the sample space, instead of just those points along a hyperplane. Specifically, if the labeled sample count is too small or the statistical shift between the labeled set and the testing set is large, semi-supervised SVM's jointly utilize the density of the feature space and the label space together. Unfortunately, the problem size (projected high-dimensional kernel space) of semi-supervised SVM's is very large and thus requires both a large amount of time for optimization and resources for the problem space.

3.3.2.2 Score Smoothing and Feature Selection

In addition to constructing new classification models with relevance feedback, some approaches perform score smoothing and feature selection to improve relevance scores produced by the prediction function $f(x)$. Assuming all results reside in a single feature space, one work attempts to smooth the relevance labeled to one point in that feature space [52]. Using the information bottleneck (IB) principle, this approach computes clusters in the feature space that best preserve mutual information between the provided relevance labels and the low-level features of results. This particular work requires pseudo-relevance labels because one part of IB clustering utilizes relevance scores over a large set of results. While quite powerful in the right environment, this technique was not successful when reranking results where scenes with very diverse appearances were in the same relevance class. However, as cited by the original author, this limitation originates in the low-level features used and not the IB reranking process itself. A related work strives to fix this problem by explicitly performing feature selection with pseudo-relevance labels. Instead of using the rigorous IB framework proposed above, authors in a second work used mutual information

over features and pseudo-relevance labels to sub-select features used to learn a new SVM classifier [58]. Here, the features are semantic concept scores and subsets were utilized to obtain reranked lists with relative improvements of 15-30% MAP using search topics from TRECVID2005 and TRECVID2006. This approach is fully automatic, using only an initial result list computation, and wholly unsupervised, relying on pseudo-relevance labels alone, it does require a lengthy computation of mutual information for structural feature selection before training. Additionally, this method may fail if either the quality of initial query results is too poor or the underlying features (semantic concepts) are indiscriminate, both of which are core weaknesses shared among all methods using pseudo-relevance feedback.

3.3.2.3 Graph-based Label Propagation

A final area of reranking techniques involve graph-based label propagation, which have recently regained popularity because of their high performance and ease of implementation. Graph-based methods are built upon neighborhood information from low-level features within a given sample space. So, unlike their model-based counterparts, graphs do not explicitly learn a class-level decision boundary, which may be difficult to optimize and can lead relevant samples being incorrectly partitioned. Instead, graph-based methods depend purely on the underlying sample manifold and are often constructed with simple nearest neighbor techniques and can therefore be reused for an arbitrary number of classes. All graph-based works rely on the assumptions that the same label will exist for samples that reside within a *cluster* or a *manifold* (proximal points in the sample space). To evaluate a feedback task in a graph-based setting, an initial set of results are considered graph nodes and their similarity are graph edges. After construction, one or more user labels are propagated over the manifold to all other samples in the graph. Figure 3.3 illustrates the stages of graph evaluation from initial collection of samples into a small working set to final label propagation in the graph. This field often builds on one of two primary techniques: LGC (local and global consistency) [139] and GFHF (Gaussian field and harmonic function) [140]. Both methods are mature, but have only been applied to a few multimedia-based problems [48], [121], [71]. Two reasons for the lack of applied works are: the construction of a high-quality graph can still be difficult because of sensitivity to distance metrics

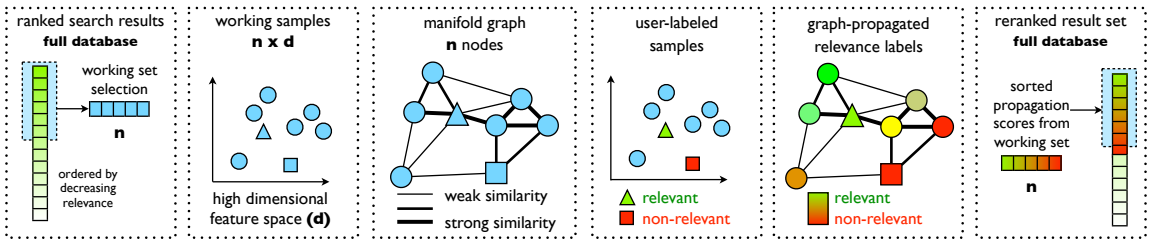


Figure 3.3: Stages of graph-based label propagation: (a) full relevance list from search, (b) collection of working samples, (c) graph construction, (d) user label assignment, (e) label propagation, (f) resorted relevance scores.

and dataset variations and that all techniques to-date require exponentially growing $O(n^2)$ affinity computations and cubically growing $O(n^3)$ matrix inverse computations.

3.4 Open Problems

Often taken for granted, there are two entities that should be actively working together in interactive multimedia search: the user and the machine. The user has a specific query or search target in mind and the machine has a vast multimedia library with a wide array of features for each indexed item. In almost all multimedia systems, users may struggle with interactive search systems due to one or more of the following problems: uninformed query formulation, a disconnection between results and navigation, the blind exploration of results, and encountering a result dead-end. While some of these problems can be mitigated with relevance feedback, traditional methods may result in a triage of unrelated results or exacerbate these problems by shifting user behavior from searching within the system to teaching it about the search target.

Uninformed Users Introduced in sec. 2.1, the easiest and most familiar expression of a user query is by textual keywords. With the use of a semantic ontology for multimedia (such as LSCOM [57]), challenges associated with text-only retrieval like sparse or missing textual descriptions are avoided. Instead, an ontology provides rich semantic descriptions of the content were embraced for search by many systems [104], [106], [16]. However, un-

less the user is familiar with the ontology, its available concepts and their specificity are unknown during query formulation. Figure 3.4 illustrates keyword and semantic concepts search methods and the potential difficulties they introduce. First, the choice of the specific

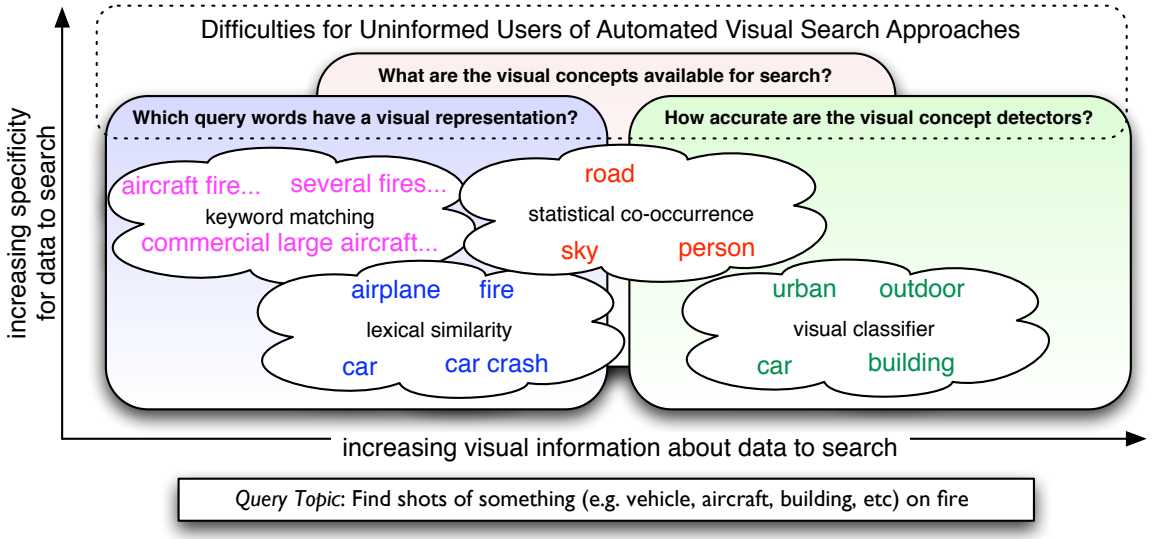


Figure 3.4: Automatic search methods and their potential difficulties for users unfamiliar with the data to search or methods available.

keyword for a search can dramatically effect the result quality of a search system. Unless intimately familiar with the content itself the user lacks information about which words would best match the content he or she is searching for (i.e. descriptions of people, places, objects, scenes, etc.). Second, if using semantic concepts trained with the procedure described in 2.2.2, non-expert users will be unfamiliar with the underlying ontology vocabulary and accuracy of each semantic concept. For example, given the query topic "find images of buildings at least six stories tall" and the choice of semantically related concepts *building* and *urban*, users may be unsure of which would be more appropriate and more accurate. As an *uninformed user* the user is burdened not only with expression of the query but also learning about the system's limitations and datasets.

Difficulty in Result Navigation Once search results are presented to the user, there is often difficulty in efficiently navigating through the system's results. Typically, search results from a multimedia database are returned in the form of a list with small thumbnails

or animated icons for inspection. Within this presentation, there is little information to help user understand why the search system returned these results, how each image is related to the different parts of the user's query (i.e. the keywords or semantic concepts), and most importantly, how to adjust the user's inspection strategy to find relevant results such as fast-skimming to cover a large quantity of results or in-depth browsing of the results. All of these difficulties arise from the *disconnection of the browsing interface and the query criterion*. In existing systems, once evaluated for a search the initial query criterion are lost and the relationship of results to the selected semantic concepts is not indicated. While some impressive visualization systems have been proposed to speed inspection of search results [125], [2] and user studies on search strategies conducted [18], new systems fail to accurately indicate the influence of and relevance to the original query criterion. Despite the potential benefits of such approaches, users may often become "puzzled" or "uninformed" about the specific criteria being used in any given point leading to an equivalent "blindspot" that will likely disengage users from continued exploration.

Blind Exploration Expert and novice users often need to reformulate a query after its evaluation and search result inspection. This reformulation process can either be to slightly refine an existing query, like adding emphasis to a specific keyword, or to dramatically change the query criterion and evaluate it anew. In both cases, users commonly enter a *blind exploration* process and fall back to searching by trial and error, as shown in figure 3.5. In this example, without knowledge about how the results of each semantic concept relate, the user must iterate through different concepts to discover those most relevant to the query topic. To preempt this time-consuming strategy the search system should utilize observations about user patterns and behaviors during previous result exploration and judiciously recommend new strategies during future query formulation and result exploration stages.

The Result Dead-end When navigation problems are taken to an extreme the user may enter a catastrophic result dead-end. A limitation enforced by traditional interactive systems is the order of result inspection. Most implementations allow only linear traversal of the result list during result inspection and require that the user refine his or her query to

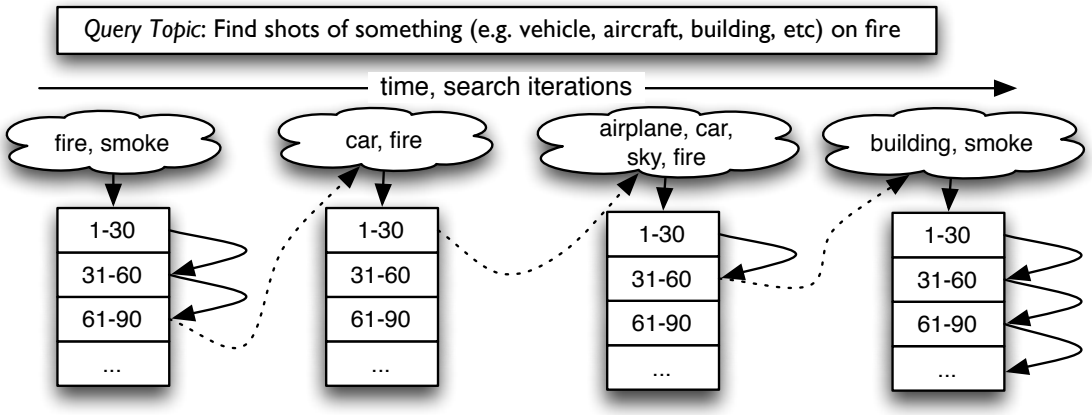
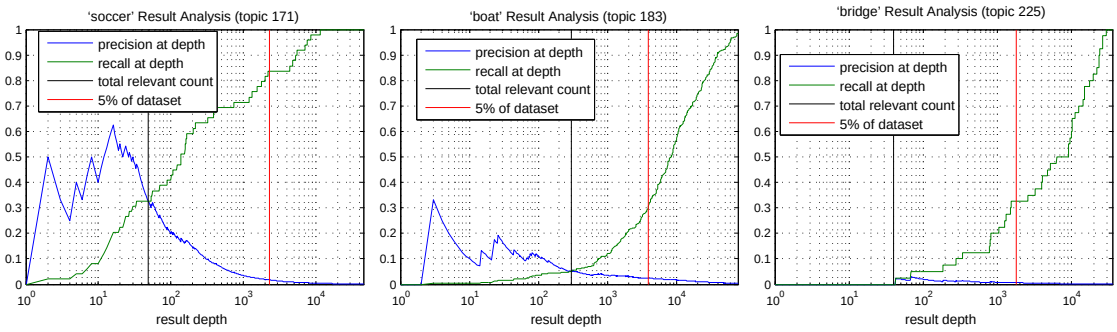


Figure 3.5: Interaction with a search system where the user resorts to trial and error query formulation to explore different parts of the data set.

break free of this course. A fixed list definition can frustrate users because items deeper in a list are generally less relevant than those that precede it, which effectively creates a *result dead end*. Providing multiple navigation alternatives, [125] actively updates several lists based on textual, visual, temporal, and semantic concept similarity to the most recently inspected result. This novel approach allows users to break away from the static results of any one query, but it simultaneously diffuses the impact of the user's formulated query. For example, if a user begins a query looking for "something on fire", it would be possible to find related images by using temporal or textual clues. However, as a user switches between the different lists, the set of results that are actively being inspected may drift further away from the original query topic such that the user no longer understands the results at all. Multiple lists allow a break from the original query, but simultaneously prevent the user from "anchoring" to a concrete and precise query target. An alternative programatic solution involves the dynamic reordering, or reranking, of results (introduced in section 2.4) and this topic is closely examined in chapter 5.

Triage of Irrelevant Results Similar to the "result dead end", the *triage of irrelevant results* naturally occurs because a user's query seldom *exactly* aligns with the user's search target. Users have become accustomed to result lists sorted by decreasing relevance to a query. Most users will inspect results at the top of lists even if they feel it is not the best answer for their search target [41]. In addition to the position of relevant items within a

list, the sheer volume of non-relevant items in most result lists can be daunting to some users. For example, for the search topic "find shots of a soccer goal taking place" a directly applicable semantic concept is available. However, due to the training conditions and diversity of appearances learned by a concept model, the result list will be clogged with non-relevant examples due to low-level feature confusion and the generality learned by the model. Figure 3.6 demonstrates this problem with empirical performances where



(a) goal in soccer match (TV05) (b) water with boats (TV06) (c) a bridge (TV08)

Figure 3.6: Analysis of result relevance producing (a) ideal, (b) degraded, and (c) unacceptable result triage conditions where relevance was computed with a single semantic concept that exactly matched each search topic.

each search topic had an exact mapping to a semantic concept. For comparison, the figure includes ideal (TRECVID2005), slightly degraded (TRECVID2006), and unacceptable (TRECVID2008) conditions for result triage. In the the *boat* and *bridge* examples, this figure illustrates that a user must examine over 1000 results to see only 10-20% of the relevant results, easily exceeding each search topic's total number of relevant results. Unfortunately, results from a semantic concept may quickly move to unacceptable conditions because search topics infrequently have a perfect semantic concept match under additional non-ideal conditions (i.e. realistic) and search targets may become increasingly complex or infrequent in the active dataset.

Teaching is not Searching In any result inspection process users may get stuck in a result dead-end or in the triage of irrelevant results. A common way to avoid these

situations is to constantly revise the user's results by reranking a working result set (figure 2.5(c)) or to issue a new query to retrieve more relevant results, which may not be ideal in real-time environments. Reranked lists and new queries may be optimized based on feedback that users have provided through interactive relevance assessments or in more passive fashions. Relevance feedback, described in section 3.3, is a process by which results are rescored according to classifier models trained with user relevance labels. These models are often constructed iteratively and are integrated with an active learning strategy that selects samples to be explicitly labeled by the user according to some pre-specified selection criterion as illustrated in figure 3.2. Regardless of the sample selection algorithm, the system can seldom truly represent a user's search target by iteratively soliciting result labels. Instead, each new relevance label is utilized in a point-wise approximation to that search target. These approximations, no longer directly connected to the user query and exploration, encourage the user to shift from a fully engaged "searching" behavior to one that attempts to "teach" the system about the search target. Although often helpful, *teaching is not searching* and these approaches can create several long-term problems for the user: the system may incorrectly exclude relevant results if trapped by a subset of relevant data, most relevance feedback systems must solicit a large number of assessments to accurately approximate a user's search target, and most importantly, the user's limited search time and attention are exhausted while teaching the machine about his or her search target instead of helping to find more relevant results or query strategies.

Chapter 4

CuZero: Guided and Real-Time Query Manipulation

In the previous chapter, automated tools for the analysis and search of multimedia videos were discussed in an interactive environment along with a collection of open problems. Most of these problems stem from the struggle of the user to fully understand how the system is searching a dataset and what the system knows about a particular dataset. One way to assuage pains from this struggle is to naturally inform the user of a system's capabilities during his or her explicit interaction. In this interaction, the machine can use many different tools (i.e. word to concept mapping, a wide array of concept recommendations, and simple navigation of the result space), all of which inform the user throughout the search process from query formulation to result navigation.

4.1 Combatting User Struggles in Interactive Search

In this chapter, solutions to three of the open problems introduced in section 3.4 are proposed in a new system framework: guided query formulation to help **uninformed users** and dynamic query adaptation to break free of **rigid visualization** interfaces and avoid the **result dead-end**. First, as the user's textual query is entered, the system assists and engages the user in query formulation by returning concept-based suggestions in section 4.2. With human participation, concept suggestions from many automatic mapping methods are

simultaneously presented to the user as the system's best answer to the user's query while allowing the human user to intelligently select the best concept for query formulation. Second, for a dramatic shift away from traditional visualization techniques, section 4.3 proposes a new interface that allows parallel evaluation and fast inspection of many minor query permutations (slight variations of query criterion). Using two intuitive displays, the user can quickly inspect a high number of query alternatives in one display and inspect the results of that alternative in another display. Third, adhering to all of these framework guidelines, a prototype system called *CuZero* was created. Real-time performance optimizations in CuZero to achieve fast, efficient, and engaging interactive are presented in section 4.4. Evaluations on example datasets in CuZero that were not previously possible are discussed in section 4.5. Finally, section 4.6 reviews related work and section 4.7 discusses future CuZero refinements.

4.2 Guided Query Formulation

Modern search systems attempt to tackle the problems above but often fail to form a truly coherent solution. In this section, a guided query formulation approach is explored using textual expression of a query, currently the fastest and most intuitive way for users to describe their search topic. However, as described in section 2.4, a typical multimedia query is composed of many different criteria like textual keywords, semantic concepts, and visual low-level features. Recent works in multimedia for automatic mapping of textual queries into pre-determined weight settings, or *query classes*, have achieved success when the class count is relatively small and the search topics are not too diverse [60], [128], [20]. Unfortunately, these automated approaches are unreliable for interactive search for two reasons. First, although empirically derived, it is unlikely that a user would easily recognize and describe a query topic with any of the automatic query classes. These works found generally similar classes like *named persons*, *named objects*, *sports, vehicles, or weather*, *animals*, and *general appearances* which utilized different combinations of feature modalities. Aiming to minimize perceived complexity and maximize user interactivity, focus is given to three basic search modalities for the formulation of new queries: textual keyword, an image or video content

example, and semantic concept. Second, the output of these automatic approaches is a fixed weighting for each of the various search modalities, which is not ideal for interactive search. For example, the query topic "find shots of something on fire" may assign a high weight to semantic concept search and a low weight to the video content example. While it would be easy for the user to change the weights arbitrarily, the process is futile without seeing how the query results are effected. This problem is avoided by populating an intuitive visual query space (continued in section 4.3) that the user can directly manipulate during query formulation.

Guided query formulation embraces the benefits of a strong automated suggestion system and the user's ability to intelligently formulate his or her query. While guided query formulation is a relatively new topic for video and multimedia retrieval systems, it is an active research topic in the information retrieval community at large. First, given the example above, one might ask whether it is beneficial for the search system to provide many automatic suggestions knowing that some of those suggestions may be inaccurate. Logically, by soliciting user responses to large numbers of potentially overlapping semantic concepts, the system is able to reduce confusion about the cognitive target search [28]. Specifically applied to multimedia search, if users could directly indicate the semantic or cognitive class for a query, the automated approaches above might be wholly sufficient for query formulation. Second, specifically addressing the problem of an uninformed user, it is valid to question whether potentially unrelated suggested concepts truly inform users. In a recent study involving the interactive search of a web page database, users found automated suggestions useful even if those suggestions weren't used in the final query because it prompted them to think about the search topic in different ways [56]. This study confirms that a symbiotic relationship between machine and human naturally develops when proposing automatic suggestions, even if the those suggestions are not ideal.

In this section, techniques to expand a set of user-provided keywords and map those query keywords to semantic concepts are explored. Utilizing lexical etymological relationships, statistical concept expansion, and data mining techniques, users are provided with a large relevant set of recommended semantic concepts selected based on their query keywords. The strength of any suggestion system is based on the breadth of its knowledge.

The proposed system relies on a large concept ontology of semantic concepts known as the *Columbia374* evaluated on the TRECVID2005 dataset [129]. Approaches described in this section are generic and could easily be evaluated with a modified set of concepts, such as those with a specific domain (consumer photos [73]), or those with additional specific entity detectors (MediaMill 500 [105]). Finally, new visualization techniques to maximize intuitiveness to all users and conveyance of system information were added during the compilation of these tools into a single interactive display. Adopting a technique used by *tag clouds*, each suggested concept is visually scaled by the product of a concept's frequency and accuracy (computed from a training dataset in section 2.2.2) to convey its utility to a user. Although their origin is debated, tag clouds are now ubiquitous in photo sharing cites [8], social communities, and general information repositories to quickly and visually indicate the relevance of a tag. Empirically, the usefulness of tag clouds for semantic concepts was shown in live evaluations of predecessor search systems [110], [16]. Figure 4.1 illustrates the various components of the proposed query formulation panel.

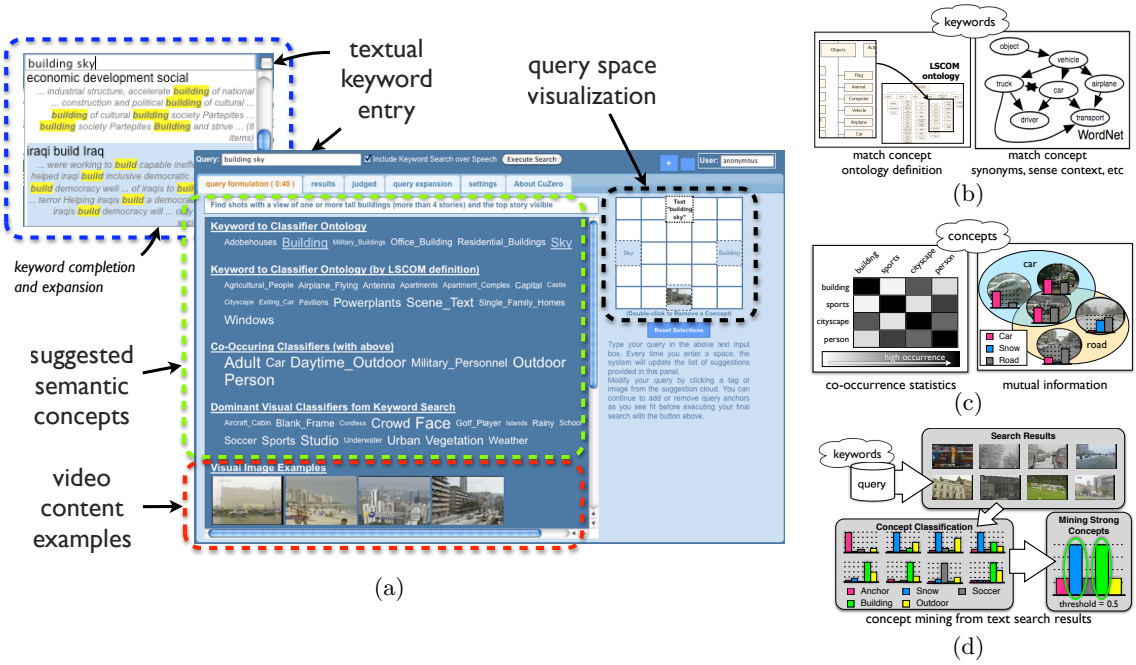


Figure 4.1: Proposed guided query formulation using the (a) query formulation panel to produce semantic concept suggestions via (b) keyword mapping, (c) statistical expansion, and (d) concept mining.

4.2.1 Keyword Completion and Expansion

Keyword completion or autocomplete is a technique used by web search engines and browsers that analyzes the letters from textual input and provides text suggestions to complete a word based on a fixed lexicon. Word uni- and bi-grams define the keyword completion lexicon and aide the user in discovering relevant speech phrases. Keyword expansion is a technique that captures completed keywords, executes a text-based search with those keywords, computes the most common keywords from the resulting documents, and presents those keywords with their context to the user. For example, the named entity "Condoleezza Rice" might be expanded to the keywords "president", "White House", and "secretary" due to relevant news stories about diplomatic efforts around the world by the former United States Secretary of State Condoleezza Rice. The steps in keyword expansion intersect larger classes of information retrieval techniques known as *query expansion*, *pseudo-relevance feedback* and *data mining*. An analog of the keyword expansion technique for video and image content is also included and discussed in section 4.2.4.

4.2.2 Lexical and Etymological Keyword Mapping

Utilizing the concept definitions from the given ontology, suggestions are discovered in two ways, as illustrated in figure 4.1(b). First, the system executes a boolean search (valid if any words match) against exact lexical matches to stemmed words [90] in names and name synonyms, as provided by WordNet, for known concepts. Second, the system performs a search for matches among stemmed definitions of all concepts. Suggestions found while matching against definitions from an ontology [57] may be less relevant than exact matches but they may be tangentially related to a topic [87]. For example, with the keyword "sports", the concept *athlete*, "a person whose job it is to perform in a sport", can be found. The quality of these matches can vary dramatically based on the specificity of definitions, so this suggestion set is visually ranked lower than direct name matches. Both of these methods assists the user to focus in on the exact ontology match that correlates to the query he or she entered. [31].

4.2.3 Statistical Concept Expansion

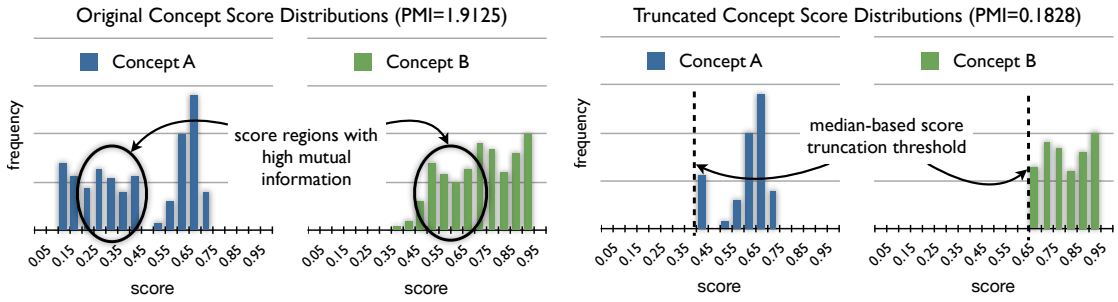
One problem introduced by the use of a large concept ontology is that users can seldom recall all available concepts. Exacerbating this problem, the semantic description of any scene can also vary dramatically according the level of detail in an ontology. Prior search systems have avoided this fact by simply verbosely listing all available concepts. However, this information overload is unlikely to help the user and does not remove the burden of having to exactly match a concept name or definition to use it. Two statistical expansion methods are employed for a given set of existing concepts, as seen in figure 4.1(c). The first method relies on a fully annotated (relevant or non-relevant) training database for each concept. In the case of the *Columbia374* this database is the TRECVID2005 development set, which contains 61901 keyframes extracted from 107 hours of 137 broadcast news programs from English, Chinese, and Arabic stations. Co-occurrence scores for every pair of concepts (X, Y) are derived from the probability of agreement in their binary labels $L = \{+1, -1\}$.

$$cooccurr(X,Y) \quad = \quad \frac{p(Y_l = 1|X_l = 1)}{p(Y_l = 1)} = \frac{p(X_l = 1|Y_l = 1)}{p(X_l = 1)} \tag{4.1}$$

While this is a reasonable solution, content from other datasets (like consumer photos, user generated web content, or surveillance videos) will likely have unique concept distributions, so the co-occurrence scores learned from the training dataset may not be valid for other datasets. To overcome this problem, a second method relying on point-wise mutual information (PMI) computed over the specific target dataset is also utilized. First, continuous concept scores computed for each dataset member are quantized into a pre-determined number of bins B, which is empirically assigned as $B = 20$ in the proposed system. Second, scores for each concept are sorted in decreasing order and the median score of each concept is determined. Each concept's score list is truncated at the median value such that no scores less than each median are included in future mutual information computations. Third, the PMI for every pair of concepts are computed with frequency of bin agreement and proportionally summed across all bins to get the total mutual information between two concepts.

$$PMI(X,Y) \quad = \quad \sum_{b_1, X_{b_1} > median(X)}^{B} \; \sum_{b_2, Y_{b_2} > median(Y)}^{B} p(X_{b_1}, Y_{b_2}) log \frac{p(X_{b_1}, Y_{b_2})}{p(X_{b_1})p(Y_{b_2})} \tag{4.2}$$

The intermediary step of score truncation prevents high mutual information scores between concepts that have a high degree of similarity for only low-scoring concepts as illustrated below in figure 4.2. Subfigure (a) demonstrates the case where unrelated concepts with similar score distribution regions can produce a high PMI score. This behavior is expected because PMI is computing bin-to-bin frequency comparisons without regard to where in the distribution two bins occur. However, the true goal of this concept expansion method is to identify more concepts with a high co-occurrence of presence (i.e. high concept scores). To emphasize this particular type of co-occurrence between two score distributions, a threshold for each concept is applied to remove low-scoring and potentially noisy concept scores. In subfigure (b), the resulting PMI demonstrates the effectiveness of score truncation in eliminating poorly aligned concept score distributions. Choice of the median as a truncation



(a) (b)

Figure 4.2: Histograms of two concept score distributions and the effect of median score truncation. Without score truncation, (a) concepts with regions of similar score distributions that are not truly correlated could be incorrectly identified as related (higher PMI). After score truncation, (b) concepts without similar high-score distributions are identified as unrelated (lower PMI).

threshold for each concept guarantees that a sufficient number of high-scoring items, the most reliable basis for mutual information computations, are preserved in the distribution because only the lower half of the original distribution is discarded. Although computationally sound, mutual information scores can also be negatively influenced by underlying poor classification scores on the target dataset as discussed. Thus, both recommendations from label co-occurence and mutual information computations are returned to the user and he or she is allowed to intelligently select relevant concepts.

4.2.4 Data Mining for Dominant Concepts

Traditional text-based or synonym-based mapping to a concept vocabulary may be unsatisfactory if there is no directly relevant concept. For example, in a query for specific named entity like "Eiffel Tower" it is possible that no concept mapping exists. While most techniques correctly map famous political figures to concepts like *politics*, *speakers*, or *heads of state*, it is unlikely that the names of buildings or locations have similarly suitable matches. Using automatically computed visual classifier scores, data mining methods can determine the most relevant concepts for a set of results. One work performs feature selection using pair-wise mutual information to select only the most useful concepts for model learning [58]. Forming the basis for CuZero's data mining, another work identifies strong correlations between concepts and results by utilizing statistically computed thresholds [136]. Both works automatically discover helpful concepts from a pre-determined set of results and their concept classifier scores. In this thesis, concept suggestions are provided using the top N results from a textual search and the scores for all concepts S. Here, the number of top results is adjusted based on the quality of text results. For datasets evaluated in this thesis, the best empirical setting was found to be $N = 40$. Next, a simple voting scheme is employed where a single vote is given for every concept $T = \{t_1, t_2, \ldots, t_k\}$ that has a score s_k above its pre-computed threshold τ_k. The threshold τ_k is the mean score of all the M top-ranked database entries for each concept. Again, M was empirically tuned to $M = 4000$ for the evaluations in this thesis. Typically, M is much smaller than the total number of samples in a dataset (commonly 5%) and an in-depth discussion of the need for score truncation is given in section 5.2.5.1. Finally, all votes are tallied and normalized by N to produce a dominance score π_k, which can be used to rank the most *dominant concepts*. To summarize, this algorithm analyzes the top results from a search and proposes that a concept is dominant if it has many high-scoring results in that list.

$$\pi_k = \frac{\sum^N count(s_{n,k} \geq \tau_k)}{N} \tag{4.3}$$

Dominant concept suggestion is useful because it may provide previously overlooked concepts because of a perceived tangential relationship (i.e. concepts *bleachers* and *benches* from the keyword "basketball"). Therefore, the data-driven nature of these suggestions

offers a valuable connection between the user and system. Unfortunately, the shortcoming of this approach is that suggested concepts depend entirely on the quality of results returned from an initial query. Without a valid set results from a textual search, there are no available results or concept scores for data mining.

Relevance-based Concept Suggestions A natural extension of concept data-mining for query formulation is data-mining using relevant results. Once a user has assigned relevance labels within the system, the semantic concept scores for those results can be used to provide additional concept suggestions. While this additional technique may seem unnecessary – why would a user need the system to describe results already deemed relevant – concept suggestions provided here can further *inform* the user and help to reduce system uncertainty about the search target. For example, with the search topic "find shots of a soccer goal being scored" the concepts *soccer* and *athlete* may be too generic and lead to many non-relevant shots of general game play. However, after the user finds a relevant result, the system can automatically suggest the additional concepts *crowd* and *walking or running*. While it is possible that these concepts may be suggested by other methods from section 4.2, the application of data-mining to only relevant results guarantees that suggested concepts are highly relevant to the user's search target.

4.3 Query Space Exploration

Guided query formulation informs the user about *what* he or she is searching, but additional steps are required to best answer *how* to search while breaking free of rigid navigation problems. Towards that effort, a query space exploration panel is introduced, allowing users to arbitrarily manipulate a visualization of the current query and immediately see its effect on results. While humans are capable of understanding weightings for different parts of a query (i.e. assigning more importance to textual search results than to semantic concept results), manipulating and visually representing these weights can be burdensome. Prior work allowed a user to visually vary query parameters with a small set of predefined setting pairs like *no faces* and *only faces* or *outdoor* and *only outdoor* [19]. While this interaction frees the user from meticulously altering numerical weights, non-expert users

may not understand the underlying meaning of this control or the control has no numerically expressible form as discussed [97]. Ideally, users should be able to strategically arrange different parts of a query allowing a better exploration of how the results of each query part interact.

This section describes methods to enable interactive exploration and instant refinement of a user's query. Using an intuitive visualization of the query in one visual panel, complex decisions about manipulating weights of different query parts and their effect on result inspection are implicitly made by the user during interaction. Additionally, with a visualized query, several methods can be utilized to directly refine and expand the query itself.

4.3.1 Creating and Manipulating a Query Space

In this thesis, the user's interaction with his or her query space is considered integral to a successful interactive search system. Avoiding a strict navigation strategy, CuZero allows the parallel exploration of query parameters without the execution of subsequent searches. During the result browsing stage, a second visual panel instantly updates with a precisely defined set of results, allowing content relationships in the query to be understood and used to further refine the query itself.

4.3.1.1 Query Parameters as Visual Anchors

With the departure from text-only interfaces, more intuitive and natural ways to manipulate user queries can be investigated. As a user constructs a query as described above, each part of that query is represented visually with an icon or *anchor*. The underlying representation of each anchor is a list of results scored by relevance to the query and sorted by decreasing relevance. This relevance has different derivations depending on the modality (image, text, concept, etc.) being compared as discussed in section 2.2. This anchor representation differs from previous works because it describes the relevance to a single query part and not a combined form of the query, allowing users to clearly understand how the system is searched, whether it is content-based image similarity utilizing low-level features or semantic concept classifiers utilizing complex learned visual models. Next, a 2D map, called the *query navigation map* is introduced to spatially organize a user's query as shown in figure

4.3. In the proposed system, users may intentionally position each anchor or allow the system to automatically position them with the shown heuristic order. The horizontal and

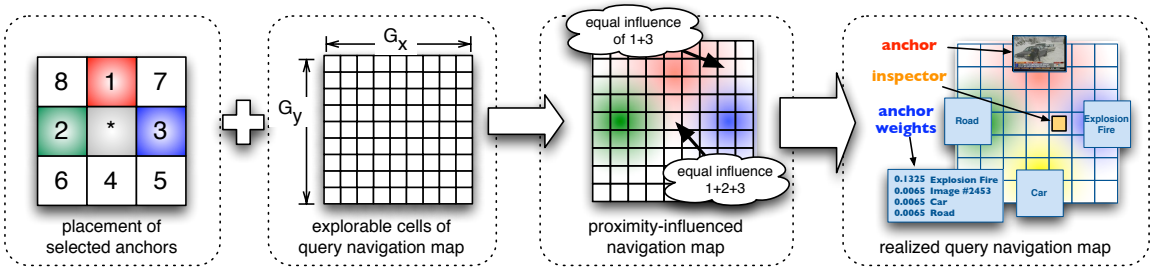


Figure 4.3: A proximity-sensitive query navigation map derived from fixed anchor placement and a spatially quantized navigation map.

vertical dimensionality ($G_x \times G_y$) of *cells* in the query navigation map can be set arbitrarily. More cells in the navigation map allow increased spatial granularity and query weighting possibilities. In typical computer-based displays, values for both variables are five or seven but they can be reduced as available display space permits.

These visualization guidelines are used in algorithm 4.1 to realize a user's query space in a rigorously defined query navigation map. First, the query anchors $A = \{a_1, ...a_k\}$ are placed and the number of cells are determined $C = \{c_1, ..., c_j\}$, the system computes the weight $\omega_{j,k}$ with respect to each anchor on a specific cell at grid position (x, y) with the algorithm below. Note that the weight $\omega_{j,k}$ is a non-linear transformation of each a cell's distance, $d_{j,k}$. In this work, the distance is Euclidean or $d_{j,k} = dist_{Euclidean}(c_i^{x,y}, a_k^{x,y})$. An additional parameter defining the weight's Gaussian decay, σ, is set to close to one ($\sigma = 0.99$) based on empirical validation. Finally, for easier comprehension by the user, anchor weights in all cells are normalized by the maximum weight for that anchor. This step guarantees that only on the exact cell that contains a specific anchor has a weight of one ($\omega_{j,k} = 1$) . This cell weight formulation results a simple query map relationship: *points closer to an anchor are more like it.* With this naturally intuitive computation, the process of delicately mixing different query anchors, regardless of their original modality, is completely hidden from the user. In the proposed system, the user is constrained to eight simultaneous query anchors, but the framework is generic enough to handle any arbitrary number. This restriction is imposed because the addition of additional anchors may decrease the intuitiveness of the

Algorithm 4.1 Anchor weighting formula for each cell in query navigation map

Input: Map cells C, query anchors A, and Gaussian re-weighting factor σ.

1. Compute the distance $d_{j,k}$ between cell position $c_j^{x,y}$ and anchor position $a_k^{x,y}$.
2. For each cell j and anchor k, compute Gaussian weight, $\omega_{j,k} = \frac{1}{\sigma\sqrt{2\pi}} exp\left(\frac{-d_{j,k}}{2\sigma^2}\right)$.
3. Normalize all cell weights by max weight for each anchor, $\omega'_{j,k} = \frac{\omega_{j,k}}{max(\omega_k)}$.
4. If any weight $\omega'_{j,k} = 1$, then $a_k^{x,y} = c_j^{x,y}$, so assign zero to other weights for cell j.
5. Derive cell priority from maximum weight, $\rho_j = max(\omega'_{j,1}, ...\omega'_{j,k})$.

query navigation map as differences between results becomes hard for the user to discern as discussed in the next section. Finally, each cell is given a priority, ρ_j, that is larger when it has a large weight for any anchor in the navigation map. Mostly used in subsequent system operations for planning and caching, this cell priority assumes that users will first navigate to cells near anchors. If this assumption does not match observed user, it can be optimally tuned during this weight computation stage.

4.3.1.2 Relevance Planning from Query Anchor Relationships

The need to quickly inspect and combine different query parameters can now be satisfied by visualizing those parameters as anchors on a 2D map. In a process called *planning*, image results for different parameter combinations are assigned to cells in the map using fusion techniques introduced in section 2.4. Recall that linear fusion is a process that uses a set of k modality weights to compute cumulative relevance of a result x_n. In each cell, multiple modalities contribute to a single relevance score $Rel(x_n)$ for each result, allowing many parameter relationships to be explored within the query space. Using algorithm 4.1, normalized weights for each anchor in the navigation map $\omega'_{j,k}$ are utilized in equation 2.9 to produce an anchor-weighted fusion of results within a cell.

$$Rel_{fused}(c_j; X) = \sum^{N} \sum^{K} \omega'_{j,k} Rel(x_{n,k}) \tag{4.4}$$

Contrary to previous search systems, for a user to see a wide breadth of query permutations, a search need only be executed and planned once with this formulation. For example, using the query in figure 4.3, a user could move the query inspector horizontally between

the concepts *road* and *explosion or fire*. The browsed results would update to show cars in these dramatically different scenes. Similarly, moving vertically between the concept anchor for *car* and the image example and the results for each cell would vary to show the effect of looking for images similar to that specific specific car versus a general car concept. Due to the Gaussian decay applied to each anchor, the user can even inspect results as if an anchor was removed from the query space (i.e. the relevance scores for that anchor have no impact on the results) by moving the inspector far away from that anchor. All of these movements in the query space represent underlying permutations of the anchor's fusion weights. Fortunately, the system knows the exact positions of anchors and dimensions of the navigation map so relevance scores for most permutations can be planned in a single batch process and be delivered to the interface without additional latency. Boolean anchor filtering, illustrated in 2.5(b), and non-linear fusions are not currently available in this formulation, but is considered one possibility for future work discussed in section 4.7. Finally, on platforms with limited computation resources, optimization strategies can minimize required planning operations by monitoring user modifications to the navigation map when replacing or moving query anchors, as discussed in 4.4.2.

4.3.1.3 Suppressing Repeated Exposures

One effect of the query anchor fusion proposed above is that the same results are repeated in multiple cells with various relevance scores. While this effect can help a user to understand contrast between anchors by showing persistent results that have high relevance scores for multiple anchors, it can also decrease the total number of results that the user can inspect. The latter effect can cause a decrease in search performance (a lower recall) and frustrate users when results from non-relevant anchors dominate many cells in the query navigation map. Looking empirically at the behavior of users with regard to visual ranking, other work utilized computer vision-based techniques to observe the gaze position of users when searching for Web documents [41]. This study's experiments found that even if a result had low-relevance to the user's search target, if it had a top ranking in the result list, users would often inspect it more closely. Returning to the query navigation map, while the system can not easily determine if results are non-relevant for a specific navigation cell, with repeat

suppression it can reduce the opportunities of the misleading the user by guaranteeing first page uniqueness. Fortunately, with a minor addition to the fusion algorithm, both the performance and frustration problems can be addressed. Processing cells according to their

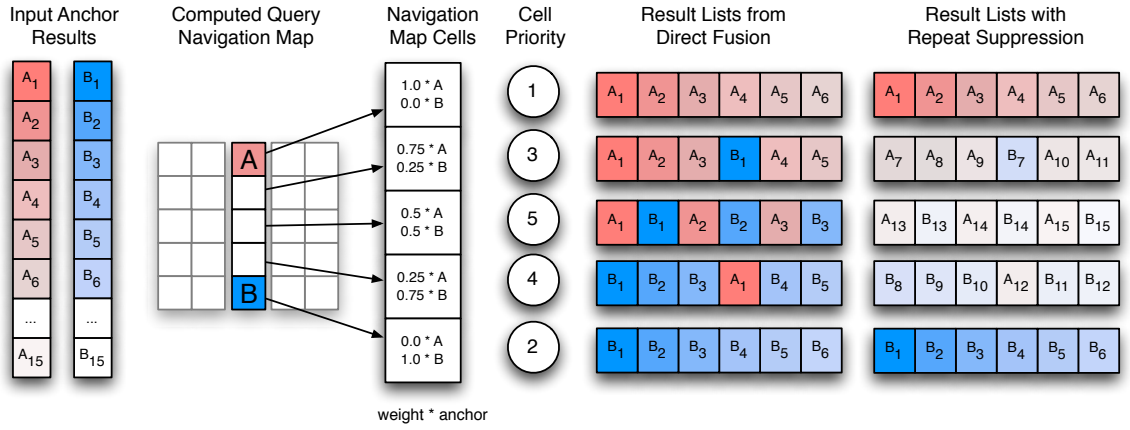


Figure 4.4: Using direct fusion to render results from multiple anchors will lead to many repeated exposures. With repeat suppression, the first visible page of results is guaranteed unique across all cells in the query navigation map. Note that top page results in lower-priority cells skip results already used in higher-priority cells.

priority, ρ_j, a total of P results for the first page of each cell can be greedily consumed from all available results. Thus, when using equation 4.5.3.2 to compute relevance scores for a cell, a result that has already been included in the top P of another cell is omitted from the lower priority cell while it does not have P unique results. After the P-th position is filled for a particular cell, relevance computation continues as usual (without suppressing repeats) so that deep exploration by the user still includes results in order of decreasing relevance.

4.3.2 Refining and Expanding a Query Space

If a search topic is particularly difficult for a user or no good query can be formulated, the number of results to inspect may be too large. Even with limitless indexing capacity, certain search topics may require the triage of thousands of results because result relevance scores are too similar, the search target is visually too small or vague and misses detection, or the time needed to inspect each result is lengthy (i.e. reviewing a video for a specific motion).

Asserting that the retrieval performance of automatic methods is bounded, one work proposed that the best solution for interactive systems was to maximize the user's inspection throughput [46]. In an extreme browsing approach, during a fifteen minute interval, as few as one or as many as nine image results were displayed to the user each second, permitting an inspection of 2000-5000 image results depending on user familiarity with the system. While this technique does push the throughput of an inspection task quite high, the system did not allow the user to stray from the initially computed list. In this thesis, a number of powerful interactive strategies are used in real-time evaluations to facilitate fast inspection and relevant result discovery. Strategies discussed in this section leverage multiple indexed feature modalities the for fast metadata filtering and similarity exploration, refinement with query expansion snapshots, and browsing observations to passively apply relevance labels.

4.3.2.1 Metadata Filtering and Similarity Exploration

As discussed in section 2.4, filtering is a natural technique to reduce the size of a result set in consideration. The type of filtering used (exclusion, union, or intersection) depends heavily on the data being considered but all types aim to prune non-relevant results. For example, if a search topic is "find images of a New Year celebration" it might make sense to employ an intersection filter to retain results with a date between a week before and a week after January 1st. With this filter, even if text-, example-, or semantic concept searches return non-relevant results, the date filter would reduce the number of inspections required. In figure 4.5, the search topic "find students in regalia" is subjected to a logical filtering strategy to exclude results without faces and only allow results from certain times of day and days of the year that common for graduation ceremonies. While filtering can be performed with scores from semantic concepts or even low-level features directly, well-defined and consistent metadata is also quite precise. This precision comes from the potential for instant loss of recall with the use of any filtering strategy. Specifically, a miss (a feature was not found) from a hard exclusion or inclusion requirement is impossible to recover from and thus the recall score of available results may be diminished. Figure 4.5 also demonstrates a secondary result panel that is shown when the user double-clicks on a single image. This sub-panel allows the immediate inspection of related results from a number of different similarity

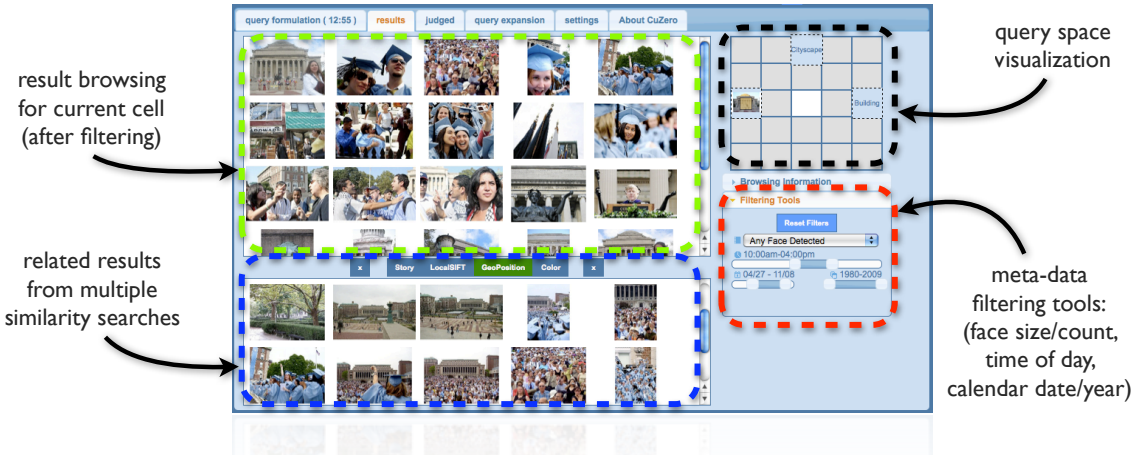


Figure 4.5: Query space navigation and result filtering interfaces in the CuZero prototype. This example shows how the similarity results for an image-based query can be filtered for the presence of faces at a certain time of day and a small range of months over many years.

evaluations. For each feature modality available in the system (including those discussed in section 2.1.2), the system displays the most relevant results. One additional feature modality available for video-based datasets is the *story*. Introduced as part of content segmentation in section 2.1.1, stories represent a higher-level syntactic boundary available in video content. Relevance scores for a story-based exploration are based on the temporal proximity of the image's source subshots and shot.

4.3.2.2 Query Expansion

When inspecting results, it is likely that the user will find a particularly promising image example, even if it does not exactly match the user's search target. Traditionally, when the user enters this state, he or she must decide whether starting a new query with this result is worthwhile. However, with the flexible query space utilized in this thesis, the user can refine their query to include the result as another query anchor and use any of its low-level features as the reference for relevance computations. This refinement process, demonstrated in figure 4.6, is referred to as *query expansion* because it adds a new anchor to the query space that was found during result exploration of an existing space. Query expansion also helps to overcome the challenge of expressing a search topic that did not include image

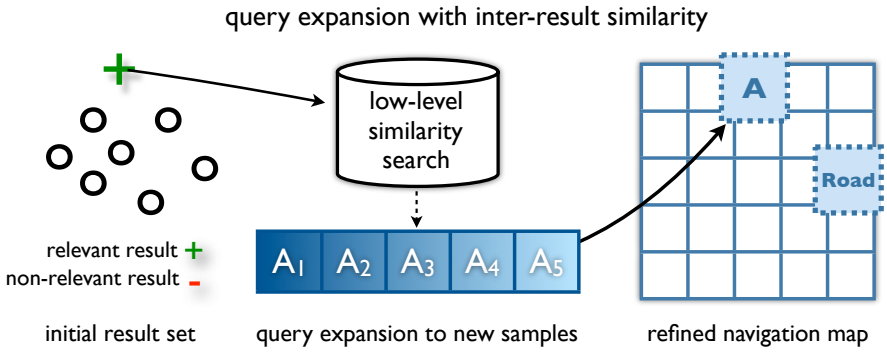


Figure 4.6: Query expansion using inter-result similarity and a refined navigation map.

examples. Examples may be unavailable if a topic is complex and difficult to describe (i.e. the interaction of many objects in several different ways) or it describes a very unique event that does not have an accurate example (i.e. a game-winning goal for a soccer match that has not yet occurred). Finally, query expansion helps to iteratively refine the query space and hone-in on the exact search target of interest. Limited only by the number of cells in the query navigation map, the user can utilize different feature modalities from numerous content examples to ideally describe his or her search target in the system. However, if carried to an extreme, the use of too many image examples can clutter the query space and combined results degrade if many marginally relevant examples are utilized. Thus, an alternate solution is to only utilize results that the user has currently marked relevant as single anchor as discussed below.

Query Expansion Snapshots A *query expansion snapshot* is a unique form of query expansion that computes relevance using *all* of the results the user has currently marked relevant. Similar to general query expansion, snapshots leverage content similarity instead of keywords or concepts to compute relevance. In the CuZero prototype, the relevance scores in a snapshot $Rel_{snapshot}(x_i)$ are weighted differences of similarity to all members of the relevant X^+ and non-relevant X^- result sets.

$$Rel_{snapshot}(x_i) = \frac{1}{|X^+|} \sum_{x' \in X^+} R(x_i, x') - \frac{\gamma}{|X^-|} \sum_{x' \in X^-} R(x_i, x') \tag{4.5}$$

In the equation above, the weighting γ is generally set to one of three values to change the influence of non-relevant results. Typical values and their interpretation are: zero to ignore

non-relevant influence, one-half for a soft penalization, and infinity for the complete exclusion of results with high similarity to non-relevant results. One benefit of query expansion snapshots formulation is the aggregation of relevance across many labeled results into a single anchor, thereby simplifying the query space and increasing user understanding. A second benefit of query expansion snapshots is the ability to compute and exactly preserve query expansion with only relevance labels available at a single point of time during the course of the search process. Over time, the user may become fatigued and forget what he or she was searching for because of overly complex interfaces or the overuse of query expansions. Either condition can cause a large drift away from the original search target. An easy way to safeguard against this behavior is to constantly "synchronize" a user's set of relevant results for the current search via query expansion snapshots, as illustrated in figure 4.7. Comparing snapshot to problems in other environments, one frequently used feature

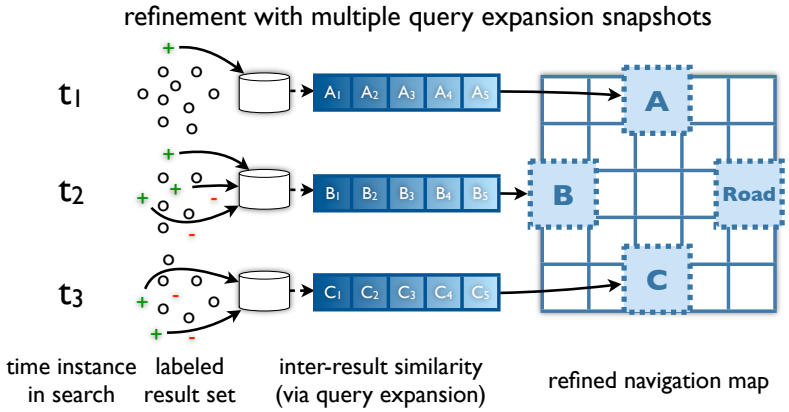


Figure 4.7: Snapshots of multiple query expansions to refine query navigation map.

is the ability to *undo* an action. For example, in word processing and database operations, there may the need to restore an environment to a previous state. While not impossible to incorporate this ability into interactive search, one must ask whether it is appropriate to revert the search environment to a prior state, knowing that a truly interactive system is both informing the user and minimizing system confusion of the user's target throughout the search process. Snapshots organized by creation time or a user-provided name, elegantly avoid the mixing of user states while also allowing the user to save and recall query anchors that strongly agree with his or her relevance labels.

4.3.2.3 Passive Browsing Observations

Methods discussed in this section rely on relevance labels explicitly provided by the user to refine and expand a query space. Passive browsing observations, however, permit the system to work with the user to gradually apply relevant and non-relevant labels to a large number of results without explicit user actions. In this work, user movements within the query navigation map create a unique opportunity to passively observe and estimate relevance from behavioral preferences. One observation from query navigation is exact knowledge of anchor weights for the cell that the user is inspecting. Through the course of a user's interaction with the system, simple statistics for sojourn time or total results inspected in one or more cells can be accumulated and correlated against the anchor weights for those cells. Assuming the user is fully engaged by the system during a search, the level of interest, and thereby anchor-specific relevance, to the user's search target can be approximated with these statistics alone.

In addition to observations from the query navigation map, the system can also apply passive or implicit labels to results that the user has inspected in the browser panel as discussed in section 3.3.1.2. In traditional search systems, results are displayed in a single grid-based list. Thus, if a user wants to see more results related to their current query he or she needs only to visually scroll through that list. Taking advantage of this deeply engrained behavior, this work gives credence to the user action of browsing and scrolling results and records which results have inspected and in what order. Carrying this assumption further, the system can assign *soft relevance labels* to the results that have been inspected. Using an entirely passive observation, the system can also assign a non-relevant label to any results that the user has *inspected and not labeled relevant*. One modern work employs passive observations but attempts to explicitly identify the area and duration of inspection for each result that a user inspects [127]. Utilizing computer vision-based techniques to track user gaze time and location, results were personalized. A prediction model for predicting relevance is constructed using the gaze location and duration for words in Web documents for TF-IDF based retrieval, rectangular image regions for low-level similarity retrieval, and cumulative region similarity in videos. In almost all cases, the user-assessed relevance of personalized results was higher than the original, non-personalized version. Work in this

thesis does not depend on gaze information, so exact performance claims for relevance assessment are deferred for future work. Initial experiments show that using soft relevance labels (like the passive observations above) boosts performance when reranking results, which could be utilized in subsequent work discussed in chapter 5.

4.4 Achieving Real-time Interaction

A prototype system called *CuZero* was implemented to embody all of the ideas proposed in previous sections. During its implementation, several critical changes were added to minimize user perceived latency. Real-time interaction is a critical tenant of modern search systems. Latency in interaction combined and response times for search executions not only effect the usability of a system but also the cognitive state of its user. Using manual information management systems and cognitive studies as a reference, one work theorized the limits for various tasks that an effective search system must embody [14]. The execution times defined in this early work closely follow the attention spans of users for different tasks in a system and have been accurate enough to serve as upper limits for computer-based systems for the past two decades. The accuracy of these numbers can be attributed to their nature: human perception and cognition and not dependence on a certain user interface or search system. Subsequent work validated these response times through real-world search scenarios with mobile device users and provided additional findings that relate to the influence of environmental stimulus, which is unique to mobile users [85]. The combined results of both studies are summarized in table 4.1. As indicated, the traditional upper limit for single-task latency has been 10 seconds, but according to the follow-up study, mobile users already need smaller response times due to environmental distractions.

The CuZero Pipeline To guarantee real-time interaction for search systems, which naturally require additional computation to support interactivity, operations must be run in parallel and asynchronously wherever possible. Implemented in the CuZero prototype, the pipeline illustrated in figure 4.8 has been constructed to meet these requirements. CuZero's processing pipeline also comfortably fits the standard distributed search system models with a single, large scale content and computation *server* and one or more light-weight, browser-

task type	timeline	user's cognitive state
perceptual processing	0.0 - 0.1 s	instant interface updates with little or no latency during interaction
immediate response	0.1 - 1.0 s	maintained, continuous thought flow; no longer working closely with the data
unit operation	1.0 - 10.0 s	maintained attentional focus on dialog with system; user expects visual feedback for task duration and may want to perform other tasks
throughput limited	4.0 - 8.0 s	maintained visual focus on system before switching to environment (mobile)
	4.0 - 15.0 s	intermittent visual focus on system from environment while checking for task completion (mobile)
	1.0 - 15.0 s	maintained visual focus on system before switching to environment (laboratory)

Table 4.1: Time limits for system interaction and corresponding user cognitive states grouped by task. Initially collated from cognitive studies on perception, this experimentally confirmed table now defines the de-facto standard for human-computer interactions.

only *clients*. For discussion, the steps in this figure are categorized into three main areas as described below and then explored more deeply in the remainder of this section.

- **Query Formulation** Query formulation takes user input in the form of textual keywords and uses a retrieval methods in the database to provide suggestions and expansions for both keywords and concepts from a known ontology. Keyword expansion utilizes keyword stems to search videos and provides potential phrases that are correlated to the words entered thus far. Concept suggestions are found by analyzing the ontology in the system using lexical cues or other information-based methods to provide more parameters for the query that the user may not know of.

- **Search Execution** Search execution embodies the algorithms that actually perform image and video retrieval, usually on a server with adequate computation resources. Searches can be executed asynchronously (while the user is still formulating the query) or serially (when the query is finalized) to retrieve a set of results for the query

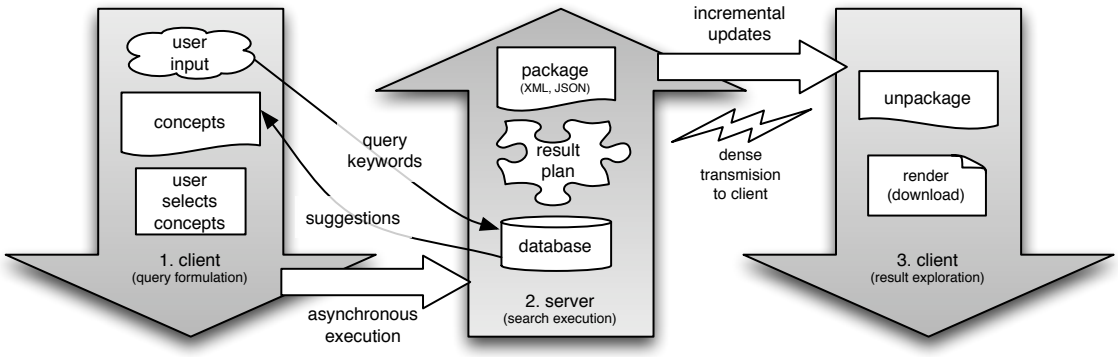


Figure 4.8: CuZero's asynchronous pipeline.

navigation map (section 4.3.1.1). After a search is executed, results for each cell of the navigation map are planned (section 4.3.1.2) and that planning is packaged into a dense data representation for transfer to the client.

- **Result Exploration** In result exploration, the user begins to navigate the query space and inspect results. Before exploration begins, the planned results must be transmitted from the server and unpackaged and images or videos must be rendered to the user's client system. Specific information for a single result (i.e. metadata, speech transcripts, etc.) are also downloaded to the client in this stage, but to optimize data transfers the download is done on-demand.

4.4.1 Query Formulation

Inspecting table 4.1, all interactions with the CuZero interface should be executed in fractions of a second to ensure that a user's experience is engaging and does not disrupt his or her thought flow. Fortunately, many of the techniques during query formulation already adhere to these limits. Additional optimizations during query formulation were created to provide immediate visual feedback to the user so that he or she is not interrupted during the perceptual processing of the search topic and interaction with the query navigation map.

Suggestions and Expansions As a user enters his or her query, the text entry component is monitored for both a maximum idle time (here 1.5 seconds) or they key-press of a

spacebar, which specifically delineates keywords. When either trigger is found, the client executes asynchronous requests to map the user's query to concepts suggestions and to provide keyword completion and expansion options, as defined in section 4.2. These requests are made asynchronously so that they can be restarted or aborted if the user performs trigger actions.

Cell Weights and Planning On-Click During query formulation, users are presented with concept suggestions (illustrated in figure 4.1). When the user selects (or clicks) a semantic concept, it is immediately added to the navigation map and the initiates a request to re-evaluate the query. On the server, results from each query anchor are locally cached so only unique anchors are evaluated. Afterwards, the server proceeds to re-plan cell relevance scores and transmit results for re-rendering, which can be done in a batch mode or incrementally for changed cells, as described in section 4.4.2. Additionally, once the client receives a planning update for a single cell, it begins to render the images for that cell into a hidden page. This rendering is also performed incrementally and at a rate that typically goes unnoticed by the user in terms of responsiveness of the client interface and transmission resources required. This technique is highly effective because users are commonly idle for several seconds while carefully formulating their query from automated concept suggestions instead of directly executing a new query with only a few anchors.

4.4.2 Search Execution

Fast search execution is essential to keep the user engaged with the system. CuZero allows a user to manipulate a query by moving a representative parameter (i.e. a query anchor) anywhere in the navigation space. Repetitive search operations are avoided when the user moves query anchors by caching each search operation on the server so that the results are immediately available for subsequent planning and packaging operations. Other real-time optimizations include a mode that only updates the query navigation map on the user's client after significant changes and provides a multi-resolution solution for low-resource platforms.

Incremental Processing Optimizations CuZero allows users to change the query navigation map during formulation at-will; these changes may include add, remove, or rearrange operations. While this freedom is crucial to query formulation, the time cost for re-executing searches, re-planning result lists for the navigation map, and re-rendering can be prohibitive if performed for every navigation map change or for thousands of users simultaneously. Instead, CuZero monitors user changes to the navigation map in the formulation panel and computes new cell weights but the planning and rendering stages are selectively executed for each cell, c_j, if its cumulative weight change, δ_j

$$\delta_j = \sum^{K} |\omega_{j,k}^{new} - \omega_{j,k}| \tag{4.6}$$

exceeds a pre-determined stability threshold. This threshold is heuristically chosen to guarantee that the user's modification did not significantly modify the previously planned and rendered query navigation map layout.

Multi-Resolution Updates for Low-Resource Platforms In addition to minimizing resources required for the query formulation stage, some platforms may have one or more of these limitations: low bandwidth, low computation power, or limited storage resource. As the distinction between classic computing resources and mobile services is continually blurred, efforts have been made to easily allow CuZero to run on next-generation platforms. Due to throughput and computation limitations, it may be infeasible to quickly transfer a large amount of dynamic (or dense) data onto a low-resource client. For example, although mobile networks are constantly being improved, the transmission of a full planning matrix may take too long and could degrade the user experience. Similarly, to maintain a small resource footprint that is also malleable to user needs, it is also impractical to pre-cache a large amount of data, as required by the rendering stage. Thus, an alternate planning and rendering algorithm was created that allows a multi-resolution update of the user's query navigation map in such a manner that the user can immediately begin result inspection after only a portion of the planned navigation map has been transferred. As shown in figure 4.9, multi-resolution updates require that a single navigation cell be planned only once for the lifetime of the query providing a prioritized, but immediately usable navigation space. In this figure, first the results for the center cell, 'a' are transmitted from server to

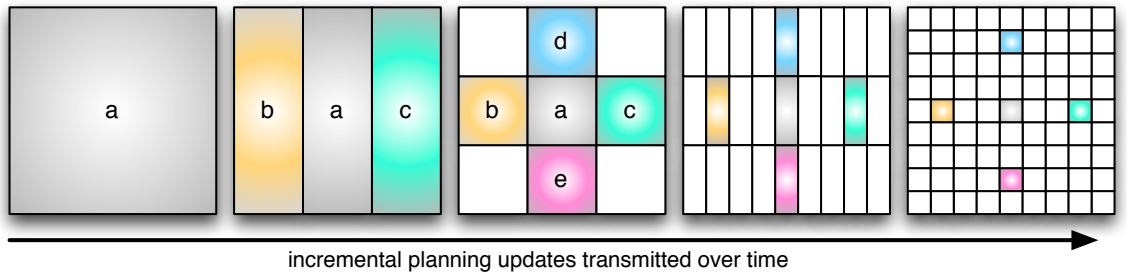


Figure 4.9: The evolution of a multi-resolution update, allowing immediate access to the navigation map even in low bandwidth environments.

client. Next, the results for cells 'b' and 'c' are transmitted to the client thereby augmenting the client's spatial resolution of the navigation map. The process repeats where the server keeps a record of which cells have been transmitted and continually partitions those cells into equal pieces until all results are delivered. With this algorithm, the transmission of results for only a few cells at a time requires much lower instantaneous bandwidth but still offers the user an instant browsing experience and a realistic navigation map that gradually increases to full resolution.

4.4.3 Result Browsing

Rendering results to the user's client system is the most time-consuming process because it requires that a large number of images or videos are downloaded and locally cached in the client. Without going to extreme measures to cache all possible results, this system delay is unavoidable so real-time optimizations in this section attempt to predict user actions and give priority to responses to those actions. Additionally, facilitating the user's interaction with all available data to quickly find relevant results, several interface-based actions are performed only on-demand.

Priority-based Planning and Rendering While computing anchor weights for each cell in the navigation map in algorithm 4.1, a priority score, ρ_j, is also derived. CuZero plans, transmits, and renders each cell according to its priority. After search execution, if a user tries to explore a cell that has not yet been rendered, the system will halt other caching operations and immediately render that cell. An adjustable time-out counter is

implemented to accommodate clients that have multiple processor cores and to prevent the client from being overloaded if the user erratically jumps between un-rendered cells.

Intuitive Interactions Common actions like labeling a result relevant or non-relevant, reviewing its metadata (or speech transcripts), or inspecting a multimedia animation (or non-static form) are all forms of click-free operations. Users can quickly inspect and label image results by hovering over an image anywhere in the interface allowing for fast interaction with the system without actively focusing on a keystroke or clicking within a small visual region. Additionally, both the query navigation map (figure 4.3) and the result panel (figure 4.10) are simultaneously displayed so there is no attentional break when the user switches between breadth- and depth-based explorations of the results at hand.

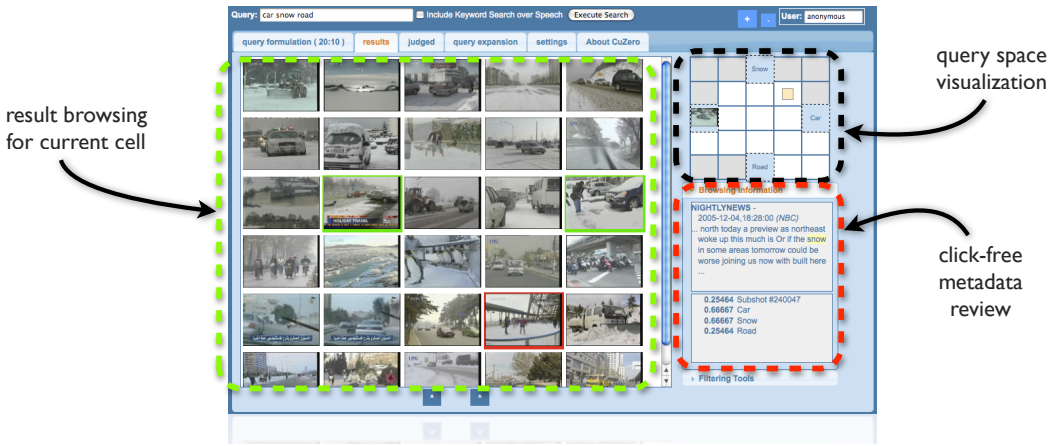


Figure 4.10: Examples of real-time interfaces implemented in the CuZero prototype: query manipulation and result browsing page. Various caching strategies are used to achieve instant responses in both interaction modes.

Continuous Scroll Rendering To minimize rendering time and resources used, CuZero does not cache all possible image results for each cell in the query navigation map. Instead only the top results on the first visible page (a customizable parameter) are rendered and cached on the client. However, to accommodate deep browsing within a particular cell (i.e. a specific anchor weighting), CuZero employs a continuously scrolling interface that dynamically renders more results as the user inspects and depletes the current set of items

in the cache. It is important to note that cells are already planned and only rendering of unseen results is completed on-demand. Every time the interface detects a scroll operation, CuZero verifies that two additional pages of unseen results are rendered and cached in the cell's scroll buffer. This dynamic loading reduces demands on the system that the user is not interested in while guaranteeing the instant availability of results for the specific query permutation the user is inspecting.

4.4.4 Design Principles

The previous sections detailed all tasks involved in CuZero search pipeline. To put these functional requirements and optimizations into a realistic perspective, the average execution times over many tests is summarized in table 4.2 for an empirical reference of throughput bottlenecks.

task	time (s)	percentage	description
keyword assistance	2.67	6.90	keyword completion and expansion
concept suggestion	3.23 (average)	8.36	map query into concept ontology
	6.16 (mining)	15.91	map query into concept ontology
	0.31 (all others)	0.81	map query into concept ontology
search execution	3.25 (average)	8.40	average time over all modalities
	4.22 (text)	10.91	pivoted TF-IDF computation
	0.13 (image)	0.32	retrieval of most similar content examples
	5.40 (concept)	13.96	retrieval of results relevant to concept
relevance planning	0.16	0.40	assign cell relevance scores to results
packaging	0.37	0.96	summarize scores and results
transmit	0.21	0.56	download data to client (same machine)
unpacking	0.03	0.07	loading scores and results into client
rendering	28.76	74.34	downloading first-page images to client
(total)	38.69	100.00	-

Table 4.2: Profiled execution time of various CuZero pipeline tasks.

Overall, there are three guiding principles for the optimizations created in CuZero. First, whenever possible, the system should be caching data while idle. There are instances where

the user is preoccupied by cognitive tasks like query formulation and result inspection in a single cell that can be utilized by the system to execute a search or render results on the user's system. Second, where latency and responsiveness are critical, incremental operations can be utilized. In the above discussions, incremental operations were proposed for low-resource platforms and situations where the user's actions did not substantially modify the query space. Third, the tantamount requirement for an interactive system is that perceived responses to user actions should be instant. Referring back to table 4.1, for a user to have no perception of the underlying computations in a search system, each action within an interface should execute in less than a second. This is an upper time limit for which the user's thought flow is not interrupted by latency due to the system. Most image and video search systems violate this requirement and a user becomes disengaged and less willing to deeply inspect the results in those systems. CuZero was designed with each of these real-time optimizations in mind and provides the necessary status updates or visual indicators where extended delays are unavoidable. In subsequent chapters in this thesis, these goals are also adhered to so that CuZero is accessible to a wide audience of users using a variety of platforms.

4.5 Experiments

CuZero was implemented with the above proposals and evaluated over a number of datasets. This section presents observations to both quantitatively justify the proposed methods using objective evaluation metrics from TRECVID2008 as well as free-form evaluations that seek to answer challenging search topics by any means necessary.

4.5.1 Overall Analysis

TRECVID is an annual international evaluation of numerous video-based tasks. As described in section 2.3, TRECVID fosters a spirit of innovation in multimedia research and provides an objective environment for task evaluation of both industrial and academic systems. In this section, the TRECVID2008 dataset consisting of educational and documentary videos in Dutch were used to analyze the performance of an expert and novice

user across twenty-four search topics. These topics were selected and assessed by NIST as part of the TRECVID evaluation and performance is provided in terms of inferred average precision. Inferred average precision is an approximation of average precision used to reduce the amount of assessment time required for a particular task [132].

Figure 4.11 presents the performances of both an expert and novice user at different interactive stages in CuZero and by each search topic. The first subfigure above demon-

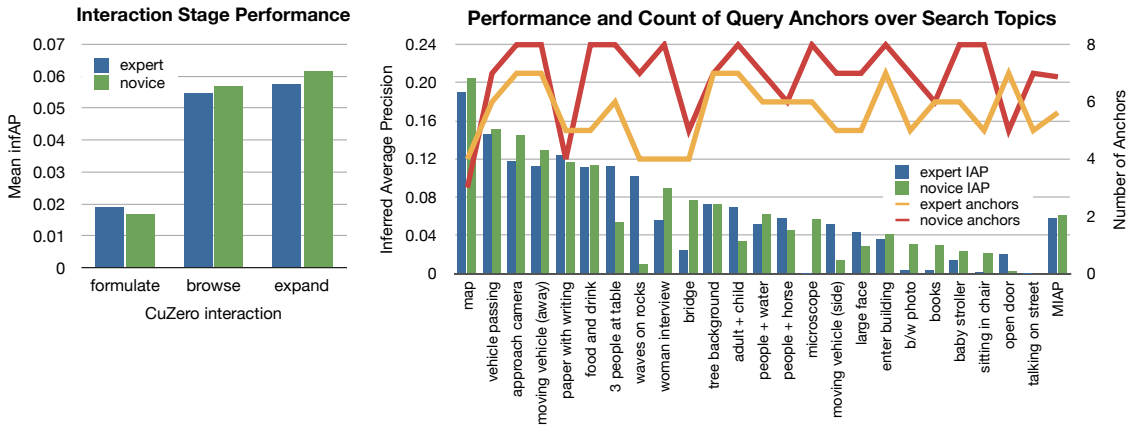


Figure 4.11: Performance of expert and novice user interactions with CuZero on TRECVID2008 search topics. Inferred average precision (IAP) is shown both at different stages of interaction with CuZero across search topics and for each search session's final results. The second figure also indicates the number of anchors used in the query navigation map during each user search session.

strates that each stage of interaction with CuZero improves search performance. Consistent performance improvement is observed starting from just the query anchors provided by the user during query formulation (section 4.2) to a full query space navigation and manipulation stage (section 4.3) to a result list from automatically expanded queries (section 4.3.2.2). It is interesting that these performance gains are also consistent for both expert and novice users. One unexpected outcome of this evaluation was the fact that the novice user regularly outperformed the expert. The second subfigure provides more insight on this topic by indicating IAP scores and the number of anchors used in each search session. Correlating these two figures, this performance disparity can be explained by an interesting phenomenon: the novice user consistently used more anchors during query formulation

and navigation. Unaware of the technical impact of adding another anchor to the query space, the novice was able to inspect more diverse sources of query relevance (i.e. more concept anchors or more image example anchors). The expert user, on the other hand, was more conservative in the choice of anchors, which lead to a higher MIAP after the first "formulation" stage of interaction, as confirmed by the first subfigure. In summary, while demonstrating CuZero's utility, this unexpected performance disparity demonstrates that the CuZero prototype is easy to understand for novice users while simultaneously leveraging the strengths of its multimodal query formulation and navigation foundations. Although not discussed here, additional informal experiments also confirmed that the passive observations from a user's search session can further improve result relevance and are a promising direction for additional work.

The results presented in this section demonstrate that the combined CuZero environment assists users in more effectively formulating and executing searches. In the following sections a closer analysis of CuZero's independent components is also provided in the objective TRECVID environment. First, a quantitative analysis of concept suggestion methods demonstrates the need for guided query formulation and then the implicit utility of the query space to simultaneously explore multiple queries is examined.

4.5.2 Concept Suggestion Analysis

At the core of CuZero's interactive query formulation, the outputs of many automatic tools are utilized to provide a richer user experience and diverse set of concept suggestions. As expected, the methods introduced in section 4.2 exhibit both strong and weak performances when measured with the diverse TRECVID2008 search topics. As a post-analysis process, *oracle performance* (IAP) was computed for each search topic and every available concept. Next, up to four concepts were recorded for each suggestion method when executed with the original text from each topic as provided by NIST evaluators - not the CuZero users. The upper limit for performance in this experiment is the oracle (or highest scoring) concept for each topic. As a baseline for maximal human performance, an expert also manually suggested four concepts for each search topic. Figure 4.12 plots the average relative performance of each suggestion method including manual selections, using oracle performance for

each topic as a maximum score reference.

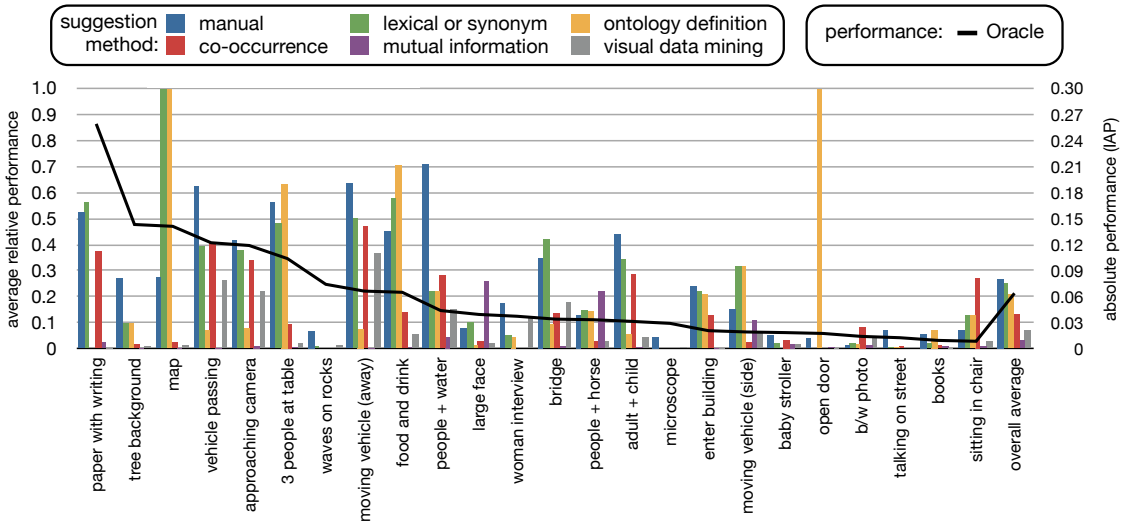


Figure 4.12: Average relative performance of concept recommendations and oracle concept scores for each search topic. The absolute performance as a black line demonstrates the best possible performance for each topic and the relative method performances, as vertical bars, demonstrate performance shortcomings of serveral methods. The varying performance levels of these methods across topics demonstrate that neither a single automatic recommendation method nor human expert is universally best and that a guided search session is required for ideal performance.

From results in this figure, four important observations can be made that all support the need for a user in the query formulation process. First, these results confirm that no one concept suggestion method is universally ideal. Correlating these results to the overall results in figure 4.11, there are instances where lexical and synonym matches or ontology definition matches are ideal like *"map"* or *"food and drink"*. This figure also demonstrates that on average, recommendations from the visual data mining method are never best and those from mutual information are infrequently so. However, in CuZero users ultimately select concepts to use from all suggestions, so it may be unwise to wholly discard these methods. Second, topics that produced weak concept recommendations from either lexical and synonym matches or ontology definition matches often produced poor recommendations with co-occurrence and mutual information expansions like *"microscope"* and *"baby*

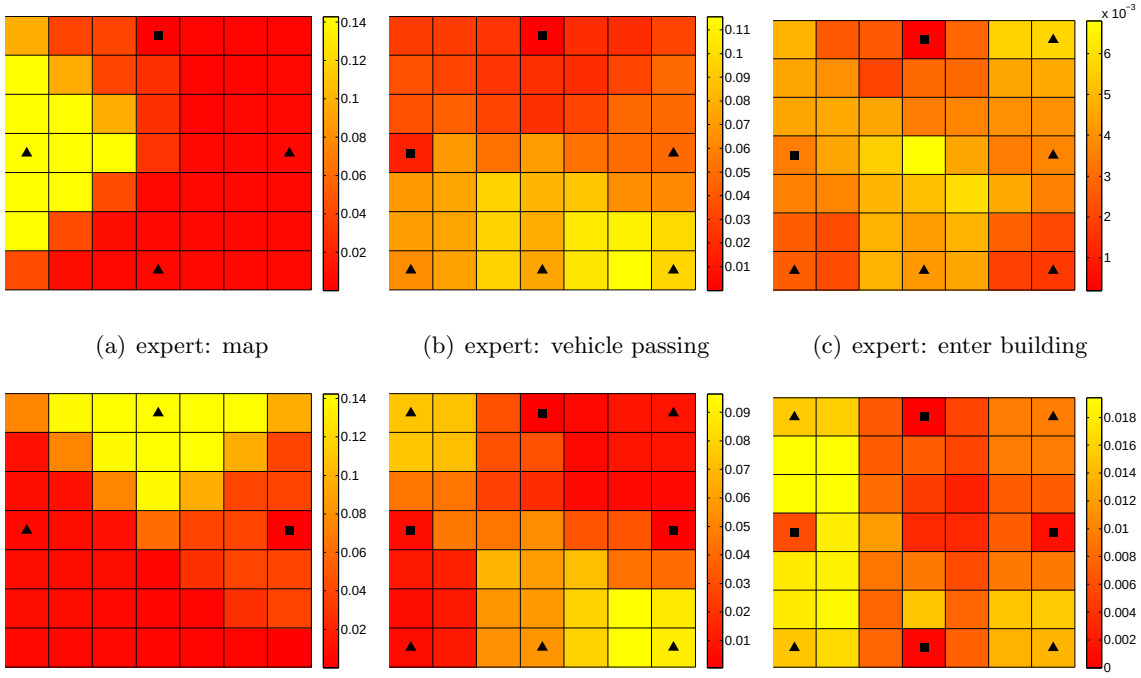
stroller". These are sensible results of concept expansion because if initial recommendations for expansion are irrelevant for a search topic, there is no way for these automated tools to correct erroneous expansions. Third, even if a search topic is correctly matched to a concept via direct matches (both lexical- and description-based), disparities between training and evaluation data for these concepts or semantic intent of the search topic can lead to poor performance like *"b/w photo"* and *"talking on street"*. Without an interactive user in the query formulation process to filter automatic recommendations, some topics are guaranteed to perform poorly. Finally, even a human expert can sometimes choose incorrect concepts to answer a search topic. In the topic *"map"*, the manual suggestions were good but the combination of too many concepts weakened the overall score. The topic *"sitting in chair"* contained sensible manual suggestions but the automatic methods using statistical knowledge about the dataset identified concepts that performed better. Each of the aforementioned automatic suggestion methods were chosen to give diverse, accurate recommendations with only textual input. However, the results in this section demonstrate several unrecoverable failure cases that could be avoided with a guided query formulation process where the user observes system responses (and limitations) and selects the best query concepts accordingly. In CuZero, concepts that seem appropriate but perform poorly for a specific search topic may still be selected by the user, but during the stage of query space exploration these poor performers can quickly be identified and avoided.

4.5.3 Query Space Analysis

The query space exploration techniques presented in this work offer a unique opportunity to simultaneously explore many different query permutations without the additional time and resources required for evaluating them individually. As discussed in the previous section, some amount of query formulation will result in poor relevance for a specific search topic. Fortunately, with the intuitive spatial separation of different parameters in the query space (section 4.3), avoiding these local pitfalls is almost trivial. Utilizing the TRECVID2008 dataset, two experiments in this section validate the the need for quickly exploring multiple query permutations and the demonstrate that 2D configuration of the query space offers sufficient coverage of underlying high-dimensional query spaces.

4.5.3.1 Simultaneous Query Permutations

Experiments in this section demonstrate that instant query space exploration is both necessary and fruitful. The exact selection and placement of query components (as anchors) created during the user sessions for TRECVID2008 were recorded and analyzed. Figure 4.13 demonstrates the performances of a few best and moderately performing search topics for analysis. In each subfigure, the lighter, yellow color indicates strong performance and the darker red color indicates poor performance. To understand the similarity in interactions between different types of anchors, concept-based anchors are illustrated with triangles and image-based anchors are illustrated with squares.



(a) expert: map (b) expert: vehicle passing (c) expert: enter building

(d) novice: map (e) novice: vehicle approaching (f) novice: woman interview

Figure 4.13: Visual performance analysis of user-formulated query spaces. Each graph illustrates the effect of combining multiple anchors in a query space for different search topic. Anchor positions are indicated with triangles (concepts) or squares (image examples) and inferred average precision is higher in yellow regions than red regions.

Several observations also arise from the inspection of this figure. First, the diversity of these query spaces demonstrates that finding the highest performing combination of concept

and image anchors is non-trivial. For example, there are the simple cases of a single anchor dominating performance (in subfigures (a) and (d)) that a user could easily select with a single query formulation and evaluation round. But there are also complex cases that require a combination of multiple anchors to find the highest performance (in subfigures (b),(c),(e),(f)) that are only feasible with the intuitive query space navigation presented here or with tenuous weight permutations by the user. Second, sometimes a combination of multiple weak anchors (in subfigures (b),(c),(f)) can produce a cell of high performance, which is an observation that may appear counter-intuitive at first. However, this finding actually supports the need and use of many semantic concepts to fully formulate a query. For example, to answer the topic "cars driving on a road" the highest performing query may contain the concepts *car*, *road*, *highway*, and *outdoor*. Actual search examples demonstrating this effect are explored in more detail in section 4.5.4. Finally, these examples demonstrate that in the CuZero prototype evaluated with the TRECVID2008 topics, semantic concepts used as anchors generally perform better than images used as anchors. This observation may indicate that the relevant results were not very similar or that the distance metric and low-level features used to represent the query image were insufficient. These potential problems are deferred for future work because they are independent of the proposed CuZero system.

4.5.3.2 Efficacy of Query Space Projections

One inherent limitation of CuZero's 2D query space exploration is that it cannot represent all permutations of a high-dimensional anchor space. This limitation is inherent to the 2D visualization of the query space, which can not simultaneously account for more than three dimensional variations.

It is important to recognize that CuZero can represent any configuration of up to three anchors because the user only cares about the *rank* of results not the absolute scores. Recall from equation that fused result scores for each cell are derived by applying a simple linear weighting scheme. Consider the scenario where the user has three anchors, A, B, and C, and the corresponding weights for any cell in the query space are ω_A, ω_B, and ω_C. To

compute the fused scores any cell, the weight combination would simply be the following.

$$Rel_{fused}(X) = \omega_A Rel(x_A) + \omega_B Rel(x_B) + \omega_C Rel(x_C)$$

If only the relative ranking of the results matter and not the actual fused score $Rel_{fused}(X)$, it is possible to normalize the weights of each anchor such that they sum to a total of one ($\sum \omega = 1$) without effecting the ranking as in the following equation.

$$Rel_{fused}(X) = \frac{\omega_A}{\sum \omega} Rel(x_A) + \frac{\omega_B}{\sum \omega} Rel(x_B) + \frac{\omega_C}{\sum \omega} Rel(x_C)$$

Although there are three weights in the above equation, there are truly only two unknown values because the sum of all weights must sum to one. Using this second condition, a simple substitution can be performed to reduce the original 3D space to only 2D with no loss of information so that one query space map represents all permutations of the given query.

$$Rel_{fused}(X) = \frac{\omega_A}{\sum \omega} Rel(x_A) + \frac{\omega_B}{\sum \omega} Rel(x_B) + (1 - \frac{\omega_A}{\sum \omega} - \frac{\omega_B}{\sum \omega}) Rel(x_C)$$

This substitution can be performed on high-dimensional spaces to reduce them from N to only $N-1$ degrees of freedom, but the current 2D implementation of CuZero's query space lacks the ability to visually the entirety of those spaces simultaneously.

Acknowledging this query space limitation in CuZero, it is important to understand how much utility (in terms of achievable performance) is lost in real-world situations. Utilizing the TRECVID2008 dataset and the manual concept suggestions obtained in section 4.5.2, another experiment was conducted comparing the performance of equal weight fusion, weighting combinations possible in the CuZero query space framework, and weighting combinations with no spatial constraints. For each search topic, four manually suggested concepts were utilized as the possible anchors and different methods were utilized to select anchor weights for use in equation 4.5.3.2. For the equal fusion method, scores for each concept anchor were averaged and evaluated. In the CuZero method, the four anchors of each topic were placed at the corners of a query space and the performance was evaluated for each cell in the query space. To find the ideal weighting, the maximum cell performance was retained from all corner-based spatial permutations of the four anchors. In

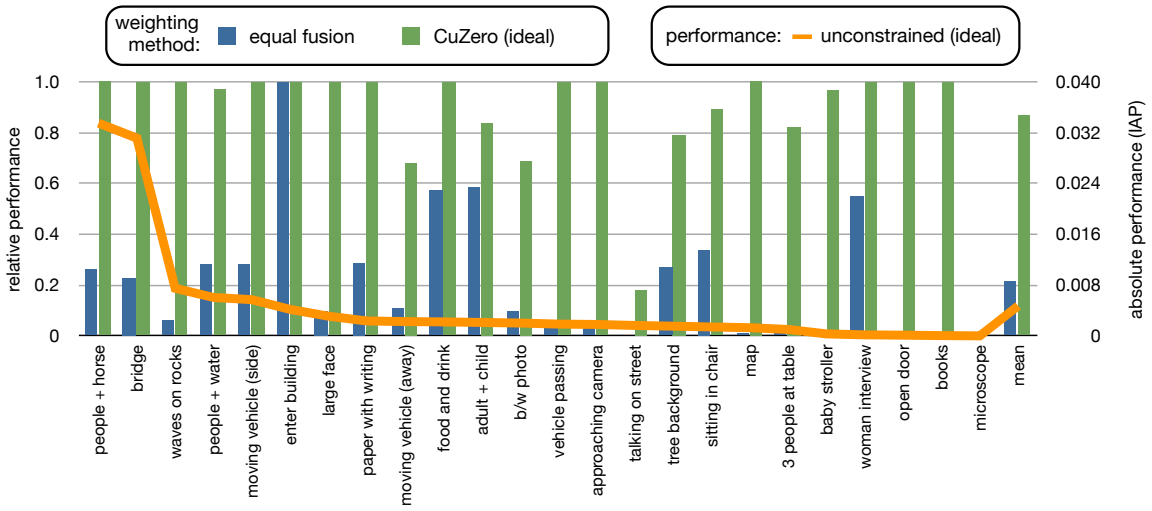


Figure 4.14: Comparison of anchor weighting methods with manually chosen concepts. Ideal unconstrained performance, indicated by the solid orange line and right-side scale, was found by searching all anchor permutations using a set of quantized weights. Similarly, ideal CuZero performance was found by evaluating all cells of possible query spaces. The best CuZero (green) and equal fusion (blue) relative performances are plotted to quantify actual performance shortcomings of these methods.

the unconstrained method, a grid search was performed where the possible anchor weights independently assume all values from the set $\omega \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$.

Figure 4.14 plots the unconstrained absolute performance (IAP) and relative performance of the fusion and CuZero methods. The figure verifies two important claims: equal weight fusion is often relatively poor and CuZero's weighting is often near ideal. In this experiment, equal weight fusion produced near-ideal performance in only one search topic. Section 4.5.2 demonstrated that finding the right set of concepts for a search topic is quite difficult, so it is not surprising that equal weighting of a set of manual concepts is non-ideal. Also, given that the concept anchors for this experiment were manually chosen by an expert, one can confirm the importance of varying weights for different concept anchors if near-optimal performance is desired. The results in this figure verifies that performance achievable with anchor placement and weightings in CuZero's query space are nearly equal to performance achievable in an unconstrained environment. Comparing specific perfor-

mance levels, CuZero achieved performance greater than or equal to 95% of unconstrained performance in 16 of 24 topics and the average performance over all topics was 87%. Additionally, comparing performance levels of CuZero and equal weighting, CuZero levels were consistently higher.

The discussion and findings in this section explored limitations of CuZero's query space with respect to anchor weighting permutations. While the query space projects the anchor space to a 2D plane, this section verified that high performance regions of those anchor spaces are possible in CuZero. It is important to note that performance values in figure 4.14 are computing with the *ranking* of the fused result scores, rather than their absolute values. This condition permits fused score normalization, performed here by dividing by the sum of all anchor weights used in the fusion process. This normalization can effectively reduces some higher-dimensional spaces (i.e. 3D or less) into a 2D query space by eliminating one degree of freedom. Future revisions of CuZero may explore non-linear projection techniques, but these alternate projection techniques must be tempered by user comprehensibility.

4.5.4 CuZero Freeform Prototype Evaluations

In an alternative evaluation setting, the goal was to test the CuZero prototype with truly challenging video search topics. These search topics included complex scene descriptions, motion requirements, and even traditional topics that most benefited from text-based approaches. These datasets in this evaluation were chosen for their size, diversity, and the general difficulty posed for interactive search. From these evaluations, the successes and remaining challenges of CuZero were analyzed as they apply to three different data domains: broadcast news, personal photo collections, and complex video surveillance.

- **TRECVID: High Quality Text and Concept Scores** The annual TRECVID evaluations, are excellent opportunities to benchmark new methods for multimedia indexing and search. In this work, broadcast news datasets (TRECVID 2005-2006) and documentary datasets (TRECVID 2007-2008) were evaluated in CuZero. In each multilingual news dataset, 120 hours of video have both relatively good textual data (from speech recognition and machine translation) and satisfactory semantic concept scores. In each documentary dataset, 166 hours of video have little textual data and

generally low- or medium-accuracy semantic concept scores.

  - "Find shots of US Vice President Dick Cheney" - textual searches returned either hit or miss results, largely from poorly aligned speech transcripts, but when combined with semantic concepts *flags* and *press conference* accuracy was dramatically improved. While face-based filtering improved result accuracy somewhat, the most fruitful actions were came from query expansions (section 4.3.2.2) and temporal navigation within a video program for related shots (section 4.3.2.1).
  - "Find shots of a bridge" - perhaps unsurprisingly, the exact semantic match *bridges* did not yield highly relevant results. However, using co-occurence and mutual-information expansion techniques additional concepts *building*, *urban scenes*, and *residential buildings* – all of which, when explored with the query navigation map, helped to find highly relevant shots of bridges in the correct visual context.

- **Flickr+Panoramio: Rich Metadata** Flickr is a website created in 2004 that supports the upload of user-submitted content with geo-tagged references. As of July 2009, Flickr hosted over 3.6 billion images from bloggers and users abroad to express their thoughts and views [124]. Unlike general web image, unique attributes of flickr content are that it is purely user-submitted content (i.e. no advertisements), usually high-quality, tagged with user-provided keywords, and often freely available for non-commercial purposes. Panoramio, created in 2005, similarly allows the upload of user content but generally with the requirement that all images include geo-positioning metadata. To assert the effectiveness of the CuZero prototype with different low-level and meta-data features, 11258 Flickr images and 1229 Panoramio images (a total 12487 images) were crawled using the tags "Columbia University" and a geo-positon at Columbia University during February 2009. Of these images, 3313 images had valid geo-position metadata, 12466 had valid time-date metadata, 20 semantic concept detectors were evaluated, and all images had one or more textual tags.

  - "Find images with students in regalia" - while, these textual tags resulted in poor textual tags for search, the data-mining approach did find the semantic

concepts *person* and *daytime outdoor*. Using these two concepts, a secondary filtering using the dates around typical graduations (June 15 - August 15) and auto-detected face information yielded high results. An example of this topic being evaluated is available in figure 4.5.

  - "Find images of the landmark San Remo" - textual results for this landmark were erroneous, but example images could be found. Once an example was found, utilizing query refinement and a geo-position filter, images around Central Park with the San Remo from different angles were correctly found.

- **YouTube: Complex Video Examples** Youtube has become a premier Web repository for user-generated web content ranging from high-quality excerpts from commercially produced content to unedited home-video uploads. To measure the effectiveness of example-only queries, CuZero was injected with a large number of long-range surveillance content. A textual search with the phrase "UAV" was executed and 61 videos were crawled for a total of 29 minutes and 4990 keyframes and evaluated with 33 semantic concept detectors.

  - "Find shots of people running near vehicles" - utilizing visual low-level global features and local object motion extracted from the example query clip, the CuZero prototype was able to correctly find images of both people running and vehicles driving. With the introduction of other semantic concepts *road* and *urban*, different city- and desert-based scenes could be identified.

In these evaluations, CuZero was tested in several capacities. First, evaluations of both fused text- and concept-based search and concept only search were verified in TRECVID datasets. These datasets provide realistic search scenarios because of the large dataset sizes and the creation of search topics derived from the data itself. Next, filtering techniques utilizing date and geo-position metadata were combined to create a pruned, high-precision set of results from a large set of loosely tagged user images. Finally, using video examples as primary query parameters, hybrid events were accurately detected and differentiated with the aide of semantic concepts.

4.6 Related Work

This chapter proposes guided query formulation and intuitive query navigation to solve the user struggles introduced in section 4.1. While combined solutions themselves are unique, the methods incorporated into each do draw from a larger pool of related work, mostly derived from automatic search approaches.

Query Formulation The basic intent of guided query formulation is to automatically inform users of alternative and potentially useful query parameters. In CuZero, keyword expansion is achieved by searching for matches within the database of automatic speech transcripts from the target video. Another keyword expansion method tries to overcome this limitation and find highly relevant named entities (i.e. people, events, locations) even in the face of poor speech recognition [20]. In this method, a corpus of external documents (i.e. online news publications) not included in the user's target dataset are collected for the same period of time as the user's target data. Next, histogram intersections are performed on the sets of documents returned from text-only search on the target and external corpora. Finally, the most frequent names of people, locations, and events that exist in both the external and target datasets are discovered and proposed as new keywords to the user. Unfortunately, the keyword expansions from this method are limited named entities and this expansion is not possible when the target data and the collected external documents occur at different times.

A second component in CuZero's query formulation panel is the suggestion of concepts from a known ontology with user provided keywords. In concept suggestion, the challenge is mapping a possibly unlimited number of keyword inputs to a smaller set of semantic concepts. Fortunately, recent work has proposed an elegant mapping of keywords even when even when they are out-of-vocabulary (OOV), or not available in the existing dataset or ontology definitions. This work uses two distinct corpora to find the most probabilistic match between the known ontology and new user keyword [126]. A new metric called "flickr-distance" is derived based on Jensen-Shannon divergence (similarity between two distributions) that constructs a latent semantic mapping from available flickr tags into a target set of semantic concepts. According to authors, the advantage of using flickr, as

opposed to other news or textual corpora, is that the keywords come from a domain with visual content. As with data mining techniques, concept suggestions from this keyword mapping process may discover concepts that the user might otherwise overlook.

In addition to text-based suggestions, other works also verified that semantic concept selection utilizing classifier-based scores generally produces higher performance. One fully automated system evaluated many of the proposed for query formulation techniques in a comparative study [79]. This study compared textual approaches such as lexical mapping, keyword expansion, and statistical expansion. Whereas the mapping and expansion methods mirrored formulations from this thesis, the statistical expansion technique proposed semantic concepts based on mutual information scores between transcript keywords and classifier scores. Authors reported a performance gain of 17% when utilizing concept classifier scores for concept expansion. Topics that involved a specific named entity experienced some performance loss because the search topic's text was already so precise. Fortunately, these findings strongly agree with the techniques included in CuZero. A second set of evaluations in this paper also propose complementary approaches to the data mining technique proposed in this thesis. First, if the search topic provides a set of visual examples, the authors propose a number of statistical tests to select the semantic concept that most agrees with the visual examples. Contrary to the data mining method in section 4.2.4 that recommends concepts for each visual example, this method attempts to align suggested concepts with the complete set of examples. While the author's method can be fruitful if the examples are fairly consistent, aggregated statistical tests may fail if the examples are diverse or produced erratic semantic classifier scores. A second technique proposed by the authors is the use of concept classifier scores to rerank a set of results. After relevance scores are computed for a set of results, a technique referred to as probabilistic local context analysis (pLCA) re-scores those results based on the agreement of classifier score distributions. Reranking a set of results with a second set of information is a powerful technique that automatically improves coherence between results. Due to its strength and importance to the search process, its application in the CuZero framework is discussed in detail in chapter 5. Although the techniques described here were originally proposed for automatic search, any technique that can add unique and query-related concept suggestions, particularly those

that utilize content properties well, can be utilized in a guided interactive environment to ease uninformed user struggles.

Query Navigation and Manipulation There are two dominant veins for query exploration: automated dimensionality reduction for navigation in the raw feature space and highly interactive user interfaces that facilitate the query space manipulation. While both approaches are quite active in the research community, few approaches have looked at the problem jointly. First, works in dimensionality reduction are often rooted in pure data analysis techniques. The application of traditional techniques like principal component analysis (PCA) for a 3D or 2D representations [50], [111], multi-dimensional scaling (MDS) [93], or self-organizing maps (SOM) [64] results in a new lower-dimensional space that a user can navigate through various interaction techniques. Beyond these classic data-analysis techniques, improvements have been proposed to supplementally organize these lower-dimensional spaces (by hierarchy [17] and clustering [107]). With all of these approaches, users often struggle to use the interactions because the results are organized in a way that is not intuitive. For example, a user may understand that images with similar color features may be grouped together, like that of a white rabbit and a snowy field. However, that same user may question why all rabbit pictures and all snow pictures are not organized together instead if he or she is actually looking for a picture of only snow, rabbits, or the combination. While the introduction of semantics into visual search is a relatively new topic, work by the author of this thesis has demonstrated that users can navigate more quickly and have a higher recall of the semantics themselves [135].

Although CuZero is the first systems to allow simultaneous query navigation and manipulation, other works solely focusing on query manipulation, or the process of changing a query after a search has been executed, have yielded promising innovations. One recent interactive search system allows query manipulation by immediately switching to an alternative query definition [25]. In this work, the image result that the user is currently inspecting is used as the input for a number of query alternatives or "threads". These threads, derived from visual similarity, temporal proximity in a video, etc. allow the user to immediately manipulate the query by changing it entirely. While this approach is fast and allows a

few query alternatives to be shown simultaneously, it can be disorienting to users because all results currently displayed will instantly change as the single result under inspection is modified. Query space manipulation need not be direct, as in the last work and this chapter. Instead, the user can provide an individual relevant and non-relevant assessment *or relevance feedback* that allows alternative projections of the feature space to be presented to the user, as illustrated in figure 2.5(c). For example, [133] proposes a method that divides the low-level feature space into multiple subspaces and applies machine learning techniques to produce new relevance assessments. Another work proposes a similar approach, but instead relying on subspace partitions alone, the system visualizes the subspace directly so the user can provide guidance on how results should next be partitioned [82] in a process called *active learning* (section 3.3.1.1). Active learning approaches have been utilized in both an offline labeling tasks [113], [4], and increasingly online in real-time search systems like [82], [75]. The revival of relevance feedback and active learning approaches has come as machine learning techniques push the possibilities of automated systems, in terms of speed and performance. However, as with the application of any fully automated approach to interactive systems, problems regarding scale (the size of the search database), required label or judgement counts (how many user assessments), and robustness to noise (erroneous labels or poor features) increasingly challenge the pace of these new approaches. Real-time query manipulation, either by directly changing query parameters or by providing relevance labels to the system is the largest distinguishing factor between the traversal of automatic search results and a truly interactive system.

4.7 Summary and Future Work

In this chapter, solutions are offered for traditional problem of uninformed users caused by user in-familiarity with the search content and available search tools and the problems of blind exploration and disconnection of query criterion. Using two unique interactive search approaches, guided query formulation and intuitive query space exploration, the user and search system work together to avoid the above problems and answer a multimedia search topic in real-time. The query space exploration approach proposed in this chapter permits

intuitive and seamless combination of any search strategies. These strategies may include traditional mutimodal approaches (similarity from visual low-level features and semantic concepts) as well as previously unavailable strategies involving the evolution of a user's query over time. At the time of writing, this approach is the first of its kind, in that it allows instant, simultaneous exploration of multiple query permutations in a single interactive environment. This chapter also demonstrated the utility of a large-scale concept ontology for both query formulation and result browsing tasks. Such ontologies have been fruitful in other works for automatic search, but the interactive exploration of independent concept scores and their relationships has seldom been closely explored. Finally, a prototype system called CuZero was created to explore these new approaches and was optimized to run in real-time in traditional client-server platforms as well as low-resource mobile platforms.

As a preliminary work, there several future directions available. First, as new multimedia content is ingested for search, an expansion of the concept ontology and inclusion of additional suggestion techniques could benefit CuZero. For example, if the concept ontology was expended to include hundreds, if not thousands, of additional concepts the likelihood of finding a relevant and useful concept to match user queries would also dramatically increase as proposed by other works [47]. Similarly, as the concept ontology is expanded, textual mapping techniques available to the user could also be expanded. One shortcoming of the existing system is that there is no method to grow the vocabulary of lexical mapping functions for new terms. For example, as new terms like "iPod" or "crowd-sourcing" are created, the existing system has no strategy in place to account for these out-of-vocabulary terms. Fortunately, recent works have begun to explore strategies for expanding a vocabulary that for multimedia and visual terms [126]. Finally, as mentioned in section 4.3.1.2, the current query exploration system can not accept non-linear or boolean fusion permutations. Referring to figure 4.3, query permutations can be computed where three anchors are fused $(A_1 + A_2 + A_3)$ but not filtered $(A_1 + A_2 - A_3)$, where the latter case excludes results with high scores for A_3. Just as anchor fusion allows the exploration of combined anchors, anchor filtering in the query navigation map allows a user quickly prune results.

In conclusion, example scenarios, possible only in CuZero were evaluated across a number of diverse and challenging datasets. These scenarios demonstrate CuZero is an ideal

platform for exploring the interaction between low-level similarity-based queries and mid-level semantic classifiers on complex data sets, like aerial surveillance or unstructured user generated videos from the web.

Chapter 5

Automatic Reranking with Relevance Feedback

In previous chapters, innovative new multimedia search strategies allowed a user to formulate and evaluate his or her query through explicit, informed interaction. However, even expert users of an interactive system with the best indexing strategies can produce results with low query relevance in the face of poor low-level features or a very large dataset.

Figure 5.1 illustrates the typical process flow when using an interactive search system. First, for a given search topic, the user is directly involved in query formulation. After formulation, the search is executed and a subset of samples from the multimedia database, called a *working set*, are given relevance scores. Second, the working set is interactively explored and inspected by the user. During exploration, the user provides relevance feedback both by explicitly labeling results and through actions performed during exploration. Third, a reranking system uses the relevance feedback provided by the user to rescore and reorder the working set of results. This reranking is automatic because the system utilizes existing user-provided information about the search target from the query and labels to rescore the working set of results. These two explicit interaction and automatic reranking stages are iterated as long as the user continues to see improvements in result relevance. In this thesis, the prototype system CuZero introduced in chapter 4 is utilized to evaluate all explicit interaction with the user.

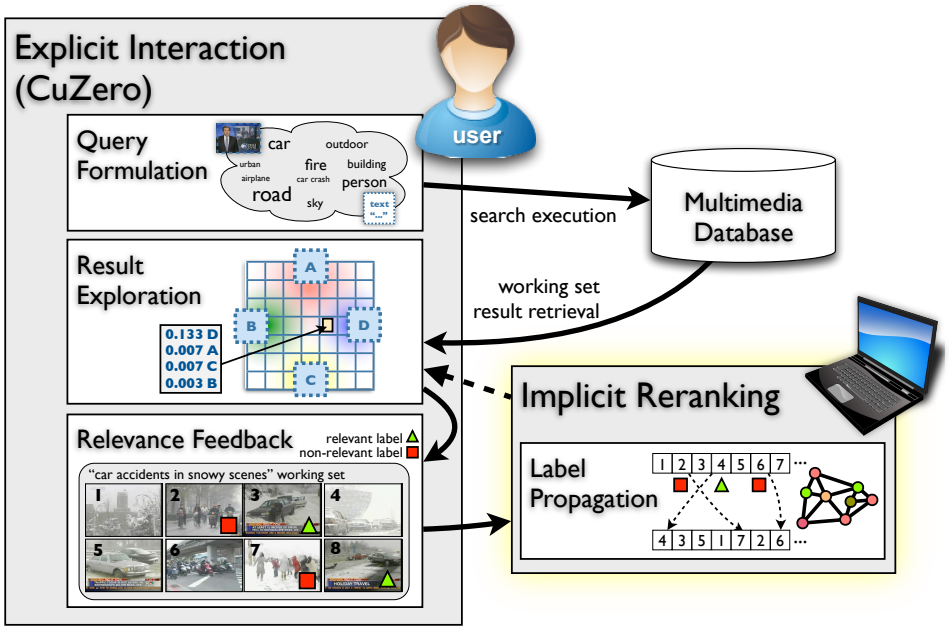


Figure 5.1: Process flow of the CuZero multimedia interactive search system. Proposed methods utilize relevance feedback from explicit interaction sessions to automatically rerank results for increased search target precision.

In this chapter, a new method for automatic reranking with relevance feedback is closely examined with the constraints of a real-time system. First, section 5.1 continues this introduction by discussing the need for reranking algorithms, existing state-of-the-art solutions for rearnking, and a brief reintroduction to graph-based label propagation. Section 5.2 introduces several techniques using *exemplar* in graphs that offer a dramatic speed-up in graph construction and evaluation while trading-off only small amounts of performance when compared to traditional graph-based formulations. Next, section 5.3 evaluates each technique on both well-known and difficult classification datasets. Finally, section 5.4 discusses related work and section 5.5 offers concluding remarks and future directions for exemplar graphs.

5.1 Recovering from Imperfect Search Results

Interactive search systems help users to define precise queries for a specific search topic. However, regardless of the skill and effort involved in query formulation, not even state-of-the art systems can retrieve results that perfectly align with the user's search target.

For example, search target misalignment may naturally occur because of differences in low-level features or variations in mid-level concept distributions between two related but not identical samples in a multimedia database. While these errors are expected and tolerated, a straightforward exploration of results may result in a long **triage of irrelevant results**, an open problem described in section 3.4. Fortunately, with additional user input (or *relevance feedback*) these errors can often be corrected to produce results more relevant to the user's search target. The remainder of this section describes a few existing solutions utilizing relevance feedback and then describes the topic of this chapter, graph-based label propagation, in more detail.

5.1.1 Modern Reranking Solutions

The goal of relevance feedback-based reranking techniques is to improve the search target precision of a predetermined set of results. Unlike general relevance feedback, the result set is implicitly defined by another search process, so the number of items in consideration is less than the entire database and the maximum recall score is fixed. Following the terminology introduced in section 3.3.2, each sample in the result set X has a real-valued feature vector x_i that is used to compute a relevance score s_i with the function $f(x)$. For any one search topic, the set's true relevance labels are Y where each sample has a binary value $y_i = \{-1, +1\}$ for any given search topic. Finally, all samples initially start in the unlabeled set X_u and are reassigned to the labeled set X_l as the user assigns relevance labels during explicit interaction.

With these formal definitions, a diverse set of reranking solutions have been created. Perhaps the longest running and most well-known field uses traditional model learning scenarios. In section 3.3.2.1, support vector machines (SVM's) are trained and evaluated in traditional supervised settings [22],[15] and semi-supervised settings [12], [38], all of which aim to maximize distance between a decision boundary and the set of labeled samples to learn new prediction functions $f(x)$. Two alternatives to constructing new relevance functions are the manipulation of the relevance scores s_i or the feature vectors x_i of the current set of samples. Discussed in section 3.3.2.2, selected works either smooth predicted scores using the information bottleneck (IB) principle [52] or perform feature selection

with point-wise mutual information. Finally, harnessing the power of underlying manifolds in a result set, recent work in graph-based label propagation [139], [140] is introduced in section 3.3.2.3. Instead of using labels to minimize decision boundary error in high-dimensional spaces, graph-based approaches use the relationships between samples to allow fast propagation of an collection of labels X_l to all other samples in the set X_u. In recent works, impressive classification performance has been achieved with the propagation of very few labels in handwritten digit, face, and concept classification experiments [71], [122].

An important property of any automatic reranking solution is that it does not disrupt the user's exploration process by soliciting for relevance labels. While many of the above solutions sufficiently rerank a set of results, often a large number of relevance labels is required. To fulfill this requirement, systems traditionally perform one reranking iteration and then ask the user to label more results to better resolve search target ambiguity. Modern works have also sought to solve this problem with the introduction of *active learning* techniques into relevance feedback. Introduced in section 3.3.1.1, an active learning system proposes results for inspection based on their estimated relevance and usefulness for the search system. Its incorporation into result inspection can resolve confusion about the most ambiguous or most similar samples depending on the respective precision and recall objectives of the user. One relevant study conducted while annotating a large dataset (over 20k images) demonstrated that employing active learning (a special type of relevance feedback) in any fashion is better than blind relevance feedback [4]. Active learning can be integrated into any relevance feedback system, so while it is out of the scope of this thesis, it can be considered an improvement for future work. Unfortunately, any system initiated action can be perceived as a distraction from the real task of result exploration and inspection can lead to lost time and possible user confusion. This open problem is strongest in interactive search environments using relevance feedback because **teaching is not searching** as introduced in section 3.4.

Works in each of the above classes have achieved success in reranking scenarios, but graph-based label propagation has consistently achieved high performance with a small number of labels. Due to these achievements, the remainder of this chapter focused on a deeper analysis of this approach and its fitness in real-time environments.

5.1.2 Real-time Limitations for Graph-based Label Propagation

While strong performance with graph-based label propagation has been shown [139], [140], little attention was given to the time requirements for these approaches. Critical to a real-time environment, both graph construction and label propagation times for large scale datasets must be considered for inclusion into an interactive system.

Among all steps in graph construction two tasks are known to have non-constant time requirements: computation of nearest neighbors and the matrix inverse for propagation. First, considering a dataset of size n, nearest neighbor computations in multimedia have been efficiently solved with a number solutions from cluster-based indexing [63] to high-dimensional hashing [114]. Respectively, these two methods require computation times of $O(log\ n)$ and $O(n^{\rho})$ where ρ is a pre-determined ratio less than one. The other large computation requirement is for an inverse of a square matrix of size n is $O(n^3)$. New techniques have been proposed that approximate inverses or speed up this computation for largely sparse matrices but they may not scale when n takes on a realistic multimedia database size on the order of 10^6.

Although graph construction can typically be performed off-line, or before a user provides labels, time required for label propagation must be included to approximate the responsiveness of an interactive system. At the time that relevance labels are provided, label propagation uses a single matrix multiplication; for k relevance classes (typically $k = 1$ in search environments), the computation time is $O(n^2k)$. Of course, some short-cuts could be taken to exclude samples from the propagation process if those samples are not among the immediate neighbors of a labeled sample [121].

Considering these time requirements, it is apparent that a compromise of performance for speed up may be necessary. The cubic inverse computation time is the most costly step in graph-based formulation and evaluation, and this time exponentially grows with the number of samples considered. Acknowledging this fact, this chapter defines an approach to reduce the number of samples n included in the inverse with an alternative graph formulation. This new approach, called *exemplar graphs*, harnesses the power of graph-based label propagation with real-time execution constraints and is the central theme for this chapter.

5.2 Relevance Feedback in Exemplar Graphs

Traditional relevance feedback methods are often plagued by execution time concerns for large datasets or potential user disruptions as a minimum number of relevance labels are collected. Recent works in graph-based label propagation have shown to be quite effective for classification tasks with few labels, but time requirements for graph construction and evaluation are not suitable for interactive environments. This section fully reviews graph-based label propagation (section 5.2.1) and introduces the concept of cluster-based exemplars as a dataset quantization technique to reduce graph construction and evaluation time (section 5.2.2). Through this approximation, the total number of nodes required in a graph can be dramatically reduced while losing little or no propagation performance in both a naïve (section 5.2.3) and closed-form (section 5.2.4) solutions.

5.2.1 Sample Manifolds in Graphs

As introduced in section 3.3.2.3, graph-based works assume that samples in the set X lying along a manifold have similar labels Y. With this assumption, an optimal prediction function F^* can be found that minimizes an objective function $\mathcal{Q}$ between user-provided labels and labels that are proposed that are propagated by the manifold itself. With this optimal prediction function, labels can be propagated from samples that have labels X_l to those that do not X_u. The LGC (local and global consistency) approach [139] is primarily used in this thesis and its formulation is discussed below because of its straight-forwardness and the ability to control the tradeoff of manifold vs. user label influence.

1. An undirected weighted graph is constructed $G = (V, E, W)$ where V is a set of vertices (the samples), E is a subset of edge weights $E \subseteq V \times V$ connecting adjacent vertices with W as a non-negative, real-valued weighting function $\mathbb{R}^+$ measuring the strength of each connection.

2. Pair-wise distances between all vertices in the graph d_{ij} are computed. The distance metric is typically Euclidean (equation 2.1) but this framework can generically accommodate alternatives. From these distances, construct an adjacency matrix $A \in \mathbb{R}^{n \times n}$

using the distances d_{ij} from only the smallest k distances for each vertex i, as symbolized by $\mathcal{N}_i$. From this step, the common annotation of k-NN is derived, which indicates a graph or list with edges only for the k nearest neighbors. The parameter σ is an empirically derived normalizer from the k-th neighbor of each sample i, $\sigma = \sum_{i=1}^{n} d(x_i, x_k^i)/n$.

$$A_{ij} = \begin{cases} \exp(-\frac{d(x_i,x_j)}{\sigma^2}), & \text{if } j \neq i, j \in \mathcal{N}_i \\ 0, & otherwise \end{cases} \tag{5.1}$$

3. With the adjacency matrix $A \in \mathbb{R}^{n \times n}$, compute the symmetric weight matrix $W = (A + A^T)$ and zero its diagonal values such that $W_{ii} = 0$. Symmetric weight matricies are increasingly standard in graph-based works because they offer increased stability and can recover incorrectly truncated edges in the graph .

4. Next, derive a square diagonal matrix $D \in \mathbb{R}^{n \times n}$ by computing the sum of each sample row and assigning it to the diagonal, $D_{i \neq j} = 0, D_{ii} = \sum_j^n W_{ij}$. With this matrix, compute the smoothness matrix $S \in \mathbb{R}^{n \times n}$, which characterizes the sample manifold as $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. Note that S is symmetric, normalized, and that self-reinforcement is avoided because the weight matrix's diagonal was first set to zero.

5. A prediction function F is estimated on the graph to minimize a cost function. Here, the cost function $\mathcal{Q}(F)$ enforces a tradeoff μ between the smoothness matrix and the fitting constraints provided by the user's labels in Y. In this implementation, each assumes a value of $Y_i = \{-1, 0, +1\}$ representing a state of non-relevant, unlabeled, or relevant with respect to the user's search target.

$$\mathcal{Q}(F) = \frac{1}{2} \left(\sum_{i,j=1}^{n} W_{ij} \| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \|^2 + \mu \sum_{i=1}^{n} \| F_i - Y_i \|^2 \right) \tag{5.2}$$

6. To minimize the cost function and obtain a closed-form solution F^*, execute the

following algebraic steps noting the substitutions $\alpha = \frac{1}{\mu+1}$ and $\beta = \frac{\mu}{\mu+1}$.

$$F^* = \underset{F \in \mathcal{F}}{\operatorname{argmin}} \mathcal{Q}(F) \tag{5.3}$$

$$\left.\frac{\delta \mathcal{Q}}{\delta \mathcal{F}}\right|_{F=F^*} = F^* - SF^* + \mu(F^* - Y) = 0$$

$$F^* - \alpha SF^* - \beta Y = 0$$

$$(I - \alpha S)F^* = \beta Y$$

$$F^* = \beta(I - \alpha S)^{-1} Y \tag{5.4}$$

7. With a unique classification function F^* pre-computed for a fixed set of samples, any new user labels can be quickly propagated through the graph by assigning non-zero values to their correct row in Y. The smothness and fitting tradeoff can be modified as necessary, where $\alpha = 0.99$ in most experiments [139].

As previously mentioned, traditional LGC is not viable for real-time environments because of the crippling cost to perform a matrix inverse in equation 5.4. For a whole dataset of 20-60k results or even a pruned working-set of as few as 2000 results in real-time environments, additional refinements are necessary if LGC is to be harnessed for automatic reranking tasks.

5.2.2 Content Exemplars

As discussed above, the largest time bottleneck in graph-based label propagation is the inverse computation of the smoothness matrix. If the original LGC algorithm is to be directly used, the most sensible solution is to reduce the size of the smoothness matrix in terms of sample count n. For this task, the focus of this discussion is turned to dataset approximation; i.e. representing a full dataset of size n with a smaller subset of size m. Once a viable approximation method has been selected it is applied to map samples to a smaller dataset. To minimize information loss during this mapping, two unique graph-based approaches are described. Finally, with these new formulations, the process for evaluating novel data (samples outside of the original graph) is introduced, an ability unavailable with classic graph-based label propagation.

5.2.2.1 Dataset Scaling with Alternative Representations

Representations that accommodate large-scale datasets have been an active research topic since the origins of signal sampling theory. Modern wireless transmission protocols, digital video compression, and even popular digital forms of audio distribution have each embraced techniques to reduce dataset requirements by adopting solutions that closely approximate original signals. In multimedia indexing and retrieval the technique in vogue may oscillate, but data reduction is a common theme: nearest-neighbor analysis [5],[131],[62], clustering and quantization [103],[63], spectral clustering [34], principle component analysis and multi-dimensional scaling [115],[64],[93], kernel-based approximation methods [30],[138], high-dimensional hashing [21], [114], and graph-based methods [139], [137], [121].

The above works demonstrate the extensive research on this topic, but few approaches address the exact problem posed in this thesis. Here, the goal of dataset scaling is to intelligently reduce the number of samples (nodes in the graph) from a dataset of X to a smaller dataset of size $\tilde{X}$. Of all of the reviewed approaches, this goal aligns very closely with that of clustering and quantization. Clustering methods directly analyze a dataset to produce groups of samples with coherent features. In information retrieval, clusters are generally organized with a tree-like hierarchy. Starting from a root node, branches are computed from two or more clusters, and leaves exist at the extremities of clusters that have no other branches. The central tenet of clustering is the use of a consistent distance metric to group samples in the feature space, but the specific metric and procedure for using raw features has seen alternative derivations [34]. The cost of traditional clustering is $O(\log n)$ because both parent and sibling nodes in the tree must be traversed for distance computation. Fortunately, recent formulations have demonstrated that distances can directly be used in organization of and in the case of bounded feature values, pre-computed hash functions can further reduce computation requirements with *vantage-point trees* [131],[62].

Exemplars have however seen uses in other machine-learning methods as "core sets" that aide both cluster formulation [5] and minimize the size of a learned model [42], [30]. The challenge in these works is determining the correct count of exemplars and the correct selection of those exemplars from the larger dataset. In this thesis, the latter problem is addressed with clustering through a vantage-point tree [131], where a fixed number of exem-

plars m are selected after clustering all members in a working set. Determining the correct count of exemplars, on the other hand, is left as a trade-off parameter for performance versus computation time. Vantage-point trees help to pick the best exemplars from a set but the number of exemplars directly effects the construction time required for graph-based label propagation. Subsequent experiments in section 5.3.3.2 vary this parameter while maintaing reasonable construction times for real-time performance. The next section continues this discussion with a closer look at quantization and its utility for content exemplars.

5.2.2.2 Exemplar Mapping

In classic signal theory, quantization seeks to approximate a large number of continuous values with a smaller set of fixed symbols; in this thesis, the goal is the same. Exemplars are samples chosen from the dataset through clustering without knowledge of their relevance to a specific search topic. Afterwards, an application of quantization, referred to as *exemplar mapping*, represents all samples in a dataset $X = \{x_1, ..., x_n\} \in \mathbb{R}^d$ with a smaller subset of those samples in an exemplar set $\tilde{X} = \{\tilde{x}_1, ..., \tilde{x}_m\} \in \mathbb{R}^d$. To fully support the graph-based formulations, two independent linear mappings are utilized to map labels between original sample set Y and the exemplar set $\tilde{Y}$. First, a matrix to map into the exemplar set is defined as $E = \{e_j \in \mathbb{R}^n, j = 1, ..., m\}$ such that

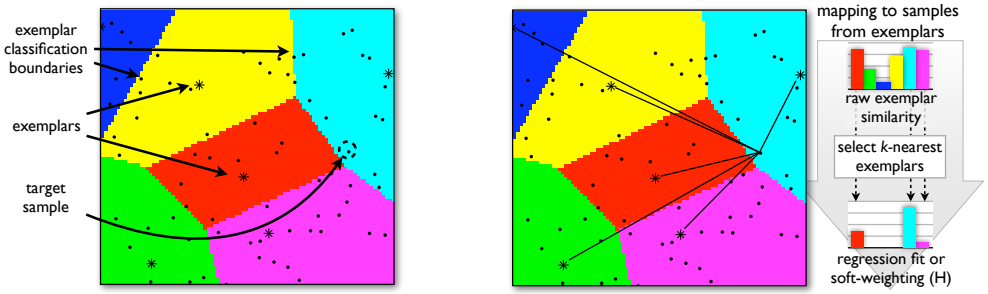
$$EY \quad \mapsto \quad \tilde{Y}. \tag{5.5}$$

A second matrix to map labels out of the exemplar set and back to the original samples is also defined as $H = \{h_i \in \mathbb{R}^m, i = 1, ..., n\}$ such that

$$H\tilde{Y} \quad \mapsto \quad Y. \tag{5.6}$$

While these two mapping matrices can be computed in a number of ways discussed below, it is important that both mappings emphasize samples in $\tilde{X}$ that are close to samples in X and not a dense mapping between all members of both sets. Figure 5.2 illustrates results of a cluster-based exemplar selection and the process of mapping a single original sample into the exemplar set using its similarity after truncation and soft-weighting.

Contrary to mappings from feature analysis methods like an Eigenvector decomposition discussed in section 5.2.2.1, the derivation of the exemplar set directly from the original



(a) toy exemplars and partitioned sample space (b) exemplar similarity and final mapping weights

Figure 5.2: Selection of exemplars as representative samples (a) and mapping out of the exemplar set to an original sample (b) from full similarity, filtered by the k-nearest neighbors to a soft-weighting scheme. In both graphics, each region's color indicates the most similar exemplar and the extent of its neighborhood.

samples minimizes information loss during two important mapping operations. First, if new samples outside of X are introduced (as is often the case with query content from the Web), they can easily be mapped into the exemplar set instead of re-computing a new Eigenvector decomposition. Second, artifacts caused by outlier samples created with an Eigenvector-based projection are avoided when mapping out from exemplars with H. Here, exemplars are actual samples selected from the original dataset and not derived from a high-dimensional embedding so there is less error during label propagation within the graph's manifold. While the impact of this second point is less obvious with discussion alone, it should be noted that the effects of mapping artifacts are greatly magnified in manifold-based score propagation scenarios.

To construct the mapping matrices, a number of solutions are available ranging from a dense mapping, to weighted quantizations, to linear regressions. First, the most straightforward solution computes distance between exemplars and original samples. With metrics defined in section 2.2.1, the transformation matrix mapping out of the exemplar space could be expressed with sample affinity as

$$H_{ij} = exp(\frac{-d(x_i, \tilde{x}_j)}{\sigma^2}) \tag{5.7}$$

where σ is the mean of all distances between samples and the corresponding exemplar. This simple approach provides a dense mapping where exemplars are represented by composites of

all samples and vice versa. This representation is problematic because the original features of an exemplar are difficult to represent with the multitude of samples. To avoid this problem, an approach similar to the graph-based approaches discussed above is adopted, where both E and H are composed of only the K nearest neighbors within X and $\tilde{X}$ respectively. This choice is a reasonable because beneficial properties of k-NN have been shown for other learning techniques [49]. In the equation below, $\mathcal{N}$ is a sorted list where $\mathcal{N}_i(k)$ is index of the k-th neighbor of the original sample x_i in the exemplar space $\tilde{X}$.

$$H_{ij} = \alpha_k \; exp(-\frac{d(x_i, \tilde{x}_j)}{\sigma^2}), j = \mathcal{N}_i(k) \tag{5.8}$$

The weighting α_k is applied to each neighbor in the list $\mathcal{N}$, and to achieve weightings equal to equation 5.7 above, all K neighbors would use a vector of ones $\{\alpha_1 = 1, ..., \alpha_k = 1\}$. A second family of solutions from previous multimedia works have applied similar non-linear mappings for sample quantizations. In recent bag-of-words (BoW) and bag-of-feature (BoF) approaches, SIFT low-level features introduced in section 2.1.2 are pre-processed with a similar quantization step [54], [88]. Pre-computed codebooks are utilized to map the high-dimensional SIFT interest points with both soft- and hard-weightings into a smaller set before being summarized in an image-level histogram. A third set of solutions acknowledges ideal sample mappings are seldom produced when directly using sample similarity from equation 5.8, and instead turns to traditional *regression* methods to appropriately map between sample sets. To empirically validate this choice, experiments in section 5.3.3.1 further explore the impact of several mapping techniques mentioned above.

In this thesis, the linear mappings are computed with a modern approach called ridge regression [44], [13], which uses a single decomposition for eigenvector analysis instead of an iterative fitting procedure. Looking specifically at H, for the purpose of regression each original sample from X is treated as the d-dimensional response V and the k nearest exemplars from $\tilde{X}$ are considered the samples U. Using this nomenclature, a linear regression problem attempts to find a function

$$f(u) = a^T u + b$$

that minimizes the residual sum of squares $RSS(a) = \sum_{i=1}^{m}(f(u_i) - v_i)^2$. Taking the derivative $\delta RSS_{ridge}(a)/\delta a = 0$, the solution is $a = (UU^T)^{-1}Uv$. When fitting the k nearest

exemplars to the d-dimensional sample the number of features is greater than the number of samples to fit. Unfortunately, this condition causes the solution to become singular and the problem is ill posed. Ridge regression therefore imposes a penalty on the norm of a with

$$RSS_{ridge}(a) = \sum_{i=1}^{m}(a^T u_i - v_i)^2 + \alpha\|a\|^2.$$

Once again, taking the derivative on the residual, the final solution is the mapping

$$a = (UU^T + \alpha I)^{-1}Uv \tag{5.9}$$

and $UU^T + \alpha I$ is no longer singular [13]. The choice of the identity matrix in the regularizer gives preference to solutions with smaller norms. It should be noted that when computing a mapping matrix, regression is performed for each sample and its k nearest neighbors. Informal experiments not included in this write-up show that performance is slightly better with this sparse regression mapping as opposed to a dense regression between all members of X and $\tilde{X}$ although computation time is slightly worse. Finally, the above discussion was provided only for the mapping H and these computations are symmetric for E.

5.2.3 Naïve Exemplar-based LGC

Recall that a closed form solution F^* was derived for graph-based label propagation. Using the properties of LGC, this solution was presented in equation 5.2 and is repeated below.

$$F^* = \beta(I - \alpha S)^{-1}Y$$

As discussed in the previous sections, the direct application of this equation on large scale datasets would be impractical for an interactive environment because of the cubic computation time required by the inverse of $S \in \mathbb{R}^{n \times n}$ in this equation. While conventional applications of LGC evaluate this equation off line [139],[121],[71], it is not practical to do so here because the samples to be used are unknown before users start the search process. Fortunately, using dataset scaling techniques discussed above, a new method called *Naïve Exemplar-based LGC* (NE-LGC) can be formulated that preserves the power of LGC while utilizing the smaller set of exemplars for all graph operations.

In all graph-based works, it is critical to guarantee that a manifold exists among nodes in a graph. Without this manifold, label propagation will be meaningless or very noisy at best. Fortunately, exemplars selected from a working set of results should still preserve sample manifolds for two reasons. First, samples in a working set are found by the best efforts available of a retrieval system. While this means that some parts of the original manifold may be lost or under-represented in the working set, the most relevant parts of that manifold should be preserved. For example, if the manifold relationships around a search target was available but weak in the original set, the working set should help to intensify those relationships by pruning unrelated (i.e. non-relevant) samples. As long as a system does not return randomly-selected results, these relationships around the search target are implicitly intensified by the similarity criterion (i.e. content similarity, semantic concept relevance, etc.) used for their retrieval. The use of a working set as a practical solution to this problem is further investigated in section 5.2.5. Second, if their number is sufficient, content exemplars derived from clustering techniques also preserve sample manifolds. While no current data reduction technique can guarantee perfect reconstruction, exemplars chosen with respect to sample density in the original dataset inherently provide support for the manifold. Subsequent experiments in section 5.3.3.3 measure the effect of exemplar count on label propagation peformance. Utilizing these assumptions, figure 5.3 demonstrates a simple adaptation of LGC that directly constructs the manifold on *exemplars* instead of samples from the working set. Here, the set of content exemplars $\tilde{X}$ selected by clustering from the

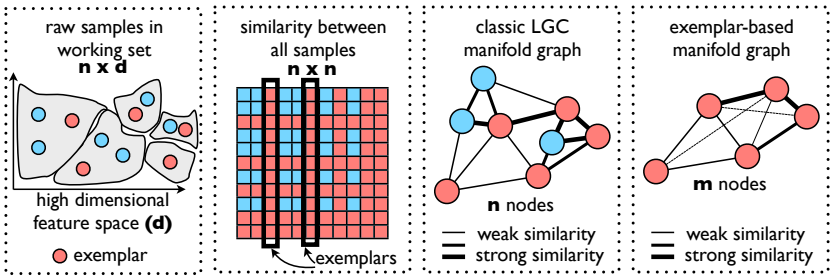


Figure 5.3: Adaptation of classic LGC, illustrated with three neighbor connections: (a) exemplar selection from raw features, (b) similarity between all samples, (c) classic LGC graph construction, (d) exemplar-based graph construction with a subset of samples.

original dataset X are directly used as vertices V in the undirected graph $G = (V, E, W)$. Thus, the adjacency matrix $A \in \mathbb{R}^{n \times n}$ computation from equation 5.1 is similarly modified to compute distance among only the exemplars themselves $A \in \mathbb{R}^{m \times m}$ as in equation 5.10.

$$A_{ij} = \begin{cases} \exp(-\frac{d(\tilde{x}_i, \tilde{x}_j)}{\sigma^2}), & \text{if } j \neq i, j \in \mathcal{N}_i \\ 0, & otherwise \end{cases} \tag{5.10}$$

Now that the function F^* operates on exemplars, it is necessary to map original samples and user-provided labels from X into the exemplar space. Fortunately, this mapping process can be readily implemented with the matrices H and E introduced in section 5.2.2. Recall that the mapping matrix allows both propagated labels to be mapped into the exemplar space (equation 5.5) and samples to be mapped out of the exemplar space (equation 5.6) similar to works using Nyström approximations [123]. The adapted form of equation 5.4 with the application of the mapping matrices to the function F^* is now provided in equation 5.11 below.

$$\begin{aligned} \tilde{F}^* = f^*(\tilde{X}) &= \beta(I - \alpha S)^{-1}\tilde{Y} \\ EY &= \tilde{Y} \\ F^* = H\tilde{F}^* &= H\beta(I - \alpha S)^{-1}EY \end{aligned} \tag{5.11}$$

One argument against NE-LGC is that the function F^* was not modified to account for the exemplar mapping process. This omission places a large burden on the ability of the matrices E and H to accurately map labels in and features out of the exemplar space and that mapping may not necessarily be symmetric. Mapping into a lower-dimensional space (i.e. the exemplar space) is analogous to sub-sampling a signal, but mapping out of the exemplar space attempts to recover lost information, which may produce overly noisy results. This second "mapping-out" process is commonly referred to as blind deconvolution and super-resolution and has been actively studied by the multimedia community for many years [6], [35], [86]. Each solution proposes a unique approach for recovering information from a low-resolution source that could be included as part of this exemplar framework, but these refinements would require additional constraints or knowledge about the signals and are deferred for future work. Unlike the blind deconvolution works, the original features

of samples in X and $\tilde{X}$ sets are always available, so accurate E and H matrices can be computed when labels are provided during the label propagation stage.

5.2.4 Exemplar-based Graph Regularization

Observing potential limitations of the NE-LGC method above, alternative graph-based approaches were also investigated. Information loss in the NE-LGC approach comes from the fact that the smoothing matrix S includes no information specific to the mapping matrices. As discussed in the last section, information loss can occur in either direction while mapping between the exemplar and original sample spaces. Fortunately this information can be better preserved by constructing an affinity matrix on the original feature space and biasing that construction with user-provided labels in the exemplar space, henceforth referred to as EGR (exemplar-based graph regularization).

Exemplar-based graphs for relevance feedback are proposed for real-time environments because of their reduction in both time and resource needs. Although ideal for speed gains, the mapping of labels into and out of the exemplar space $\tilde{\mathcal{X}}$ can result in noisier propagated labels. One way to avoid this problem, the adjacency matrix is constructed over the original sample space X such that $A \in \mathbb{R}^{n \times n}$. Using the full adjacency matrix, the original LGC objective function from 5.4, can be updated with a substitution of the Laplacian matrix $L \in \mathbb{R}^{n \times n}$ for the diagonal evaluation of the weight matrix, $L = D - W$.

$$\mathcal{Q}(F) \quad = \quad (1-\mu)\ trace(F^T L F) + \mu \sum_{i=1}^{N} \|F_i - Y_i\|^2 \tag{5.12}$$

A second way to reduce information loss from projection of a large sample space into a reduce exemplar set is to bias graph construction with label information. Specifically, the difference between sample labels Y and their label predictions F can now be divided into labeled (Y_l, F_l) and unlabeled cases (F_u, Y_u) respectively.

$$\mathcal{Q}(F) \quad = \quad (1-\mu)\ trace(F^T L F) + \mu\|F_l - Y_l\|^2 + \mu\|F_u - Y_u\|^2$$

While unlabeled samples X_u may provide information about manifold smoothness, it is undesirable to manipulate the cost function with this unlabeled information. For example, if the available unlabeled samples greatly outnumbered labeled samples, the lowest-cost F

could be biased towards neighborhood smoothness of unlabeled samples at the loss of more useful neighborhood information from the labeled samples. For this reason, all terms using unlabeled samples are dropped from subsequent formulations, as shown below.

$$\mathcal{Q}(F) \quad = \quad (1-\mu)\ trace(F^T L F) + \mu \|F_l - Y_l\|^2$$

To adapt this formulation to use exemplars, the traditional function F is replaced by the product of the mapping matrices H in an exemplar-based function G. It should be noted that for the prediction error, only rows for labeled samples H_l are used in this substitution, which correctly match elements of the mapping matrix H to the provided labels Y_l.

$$\mathcal{Q}(G) \quad = \quad (1-\mu)\ trace(H^T G^T L H G) + \mu \|H_l G - Y_l\|^2 \tag{5.13}$$

Now, this complete cost function can be minimized for a closed form solution of EGR.

$$\begin{aligned} G^* &= \underset{G \in \mathcal{G}}{\operatorname{argmin}} \mathcal{Q}(G) \\ G^* &= \left((1-\mu) H_l^T H_l + \mu H^T L H\right)^{-1} H_l^T Y_l \end{aligned} \tag{5.14}$$

The NE-LGC formulation of F^* (in equation 5.11) differs from the final EGR formulation of G^* (in equation 5.14) in two significant ways. First, graphs for NE-LGC are only be constructed once for an entire dataset. EGR graphs, on the other hand, are constructed dynamically according to new labels that are provided. Second, NE-LGC utilizes an adjacency graph constructed directly in the exemplar space $\tilde{X}$, which may encounter sampling errors. EGR, on the other hand, computes the adjacency graph on the raw sample space X and then maps that graph into the smaller exemplar space.

One recent work using prototype vector machines (PVM) [138], proposes a formulation similar to equation 5.14. This work was derived from kernel-based approximations, and not from the exemplar approach in section 5.2.2. In this work, the authors express the label function as

$$F^* \quad = \quad \left(c_1 H_l^T H_l + H^T L H + c_2 H_u^T H_u\right)^{-1} E_l Y_l. \tag{5.15}$$

In this formula both c_1 and c_2 are constants and the matrices H and E represent dense sample weightings for exemplars, called "prototypes" in the PVM work. With only a cursory

comparison, it is apparent that these two formulations are similar. The first difference is the PVM's formulation includes influence from unlabeled samples. As discussed in derivations of equation 5.13, these unlabeled samples may negatively impact the sample manifold by artificially requiring smooth neighborhoods over unlabeled samples. In PVM work, this term is often set to zero during evaluations. The second difference is that PVM expresses H and E as different approximations in formulation. However, during evaluation these two matrices are set equal to each other ($E = H^T$) in a dense mapping (i.e. all entries are non-zero) into and out of an exemplar domain. PVM authors justify a dense matrix representation by citing Nyström approximations that have little or no squared error compared to original samples [123]. In practice and the illustration in figure 5.2, sparsity is critical in the mapping matrix to reduce the influence of labels from unrelated exemplars on distant and unrelated samples. Due to this similarity, specific performance comparisons of the proposed exemplar approaches to PVM are included throughout in section 5.3.

5.2.5 Label Propagation to Novel Data

One powerful function of exemplar-based graphs is the ability to propagate labels onto any sample set, including both subsets of the original sample space and samples outside of the original set entirely. Unlike traditional LGC or GFHF formulations, the proposed NE-LGC graphs are constructed on the sampled exemplar space $\tilde{X}$ and samples are mapped into this space with the matrices H and E. EGR graphs are initially constructed on the original space X and then mapped into the exemplar space with the matrix H. Thus, to evaluate the labeling function of either the NE-LGC or EGR methods on a single sample x, one need only the corresponding rows of H and E.

$$\begin{aligned} f^*(X) = HF^* &= H\beta(I - \alpha S)^{-1}EY \\ g^*(X) = HG^* &= H\left((1-\mu)H_l^T H_l + \mu H^T L H\right)^{-1} H_l^T Y_l \end{aligned}$$

With this additional mapping step, the proposed techniques can break free of the closed dataset problem associated with by LGC and GFHF. To propagate labels onto new samples outside of the original space, only the corresponding exemplar mapping is required providing that the manifold of the original sample space X is also representative of the new samples.

This unique ability allows exemplar graphs to propagate labels to out-of-set samples beyond

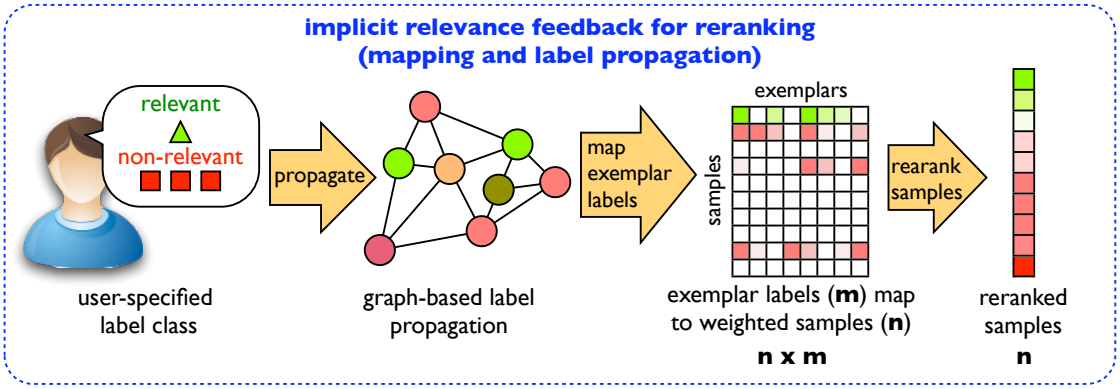


Figure 5.4: Relevance feedback for reranking with exemplars. User labels are propagated through a graph then mapped back to a set of samples for a user-specified reranking.

the initial working set without requiring additional graph construction or additional labels, as illustrated in figure 5.4.

5.2.5.1 Result Pruning for Real-time Performance

With the ability to propagate a few user labels to novel sample points, the discussion focus is returned to the issue of scale. Scale effects the utility of a multimedia search system by effecting a user's interaction experience and the ability for the system to computationally personalize that experience. In the first case, users may understand that a system has indexed millions of hours of video or all of the images on the Web, but if that system's responsiveness is bogged down (search techniques failing under stress of the dataset) or relevant result recall is low (inappropriate indexing shortcuts to accommodate scale issues), its perceived performance will be dismal. Similarly, all interactive systems strive to provide their users with the most relevant results possible. In a single search session, a system can learn how to customize results based on explicit user interaction, increasing user fulfillment and decreasing time used in the search process. To support the approaches described above, this section examines result truncation as a practical retrieval approach and its implications on a realistic search dataset.

Most relevance feedback systems will provide reranking scores for an entire dataset, but

often a more pragmatic approach may be to only rerank results in a reasonably sized *working set*. The working set of results is defined as the most-relevant subset of results currently computable given a user's specific query. A high relevance assumption is valid because the results were derived from the system's best automated retrieval process given all available information. Upon first consideration it may seem that a loss of recall is inevitable when limiting the reranking processes to a small partition of the dataset. However, such concerns can be partially alleviated by the empirical observations that with a large collection of visual concepts for search, every subshot in a dataset appears in small top lists associated with one or more concepts.

To defend against the loss of recall, it is essential to verify that reduced indexing depth still provides adequate dataset coverage for a diverse set of query formulations. In this thesis, queries are most frequently derived from semantic concept scores and low-level content similarity. Table 5.1 demonstrates that realistic video datasets from different years of TRECVID evaluations can be almost completely covered by one or more concepts and one or more feature modalities with only 5% of the indexing depth. A single sample (i.e. an

dataset (size)	**con**	**sim**	**con**	**sim**	**con**	**sim**	**con**	**sim**	**con**	**sim**
depth	0.5%		1%		2.5%		5%		10%	
TV05 (64256)	69.3	99.3	89.4	99.7	99.4	99.9	99.9	100.0	100.0	100.0
TV06 (119490)	71.8	99.7	90.5	99.9	99.3	100.0	99.9	100.0	100.0	100.0
TV07 (22084)	76.1	95.9	92.0	98.0	99.2	99.3	99.9	99.7	100.0	99.9
TV08 (112388)	27.8	99.4	87.8	99.8	98.9	99.9	99.9	100.0	100.0	100.0

Table 5.1: Precent of coverage of all subshots with concept- and similarity-based indexing over various TRECVID datasets with different indexing depths.

image) is considered covered if its position in a ranked list is lower than a specified threshold. For example, with an indexing depth of 10% and a dataset of 100 images, a coverage score of one means that each of the 100 images appears within in the top 10 results of one or more of the available indexes. Accordingly, the coverage percentages reported above utilize the *Columbia374* semantic concept scores and similarity scores from low-level grid color moment features. For coverage with similarity scores, each subshot in the database is used as a query against the remainder of its dataset.

A subshot is considered covered in a dataset even it appears only once across all indexes at a specified depth. Exploring the implications of this computation more deeply, figure 5.5 shows subshot coverage in the TRECVID2008 dataset versus the number of appearances in a concept or similarity result list. In other words, this figure illustrates the distribution of subshots across one or more truncated result lists. From this figure, one observes two

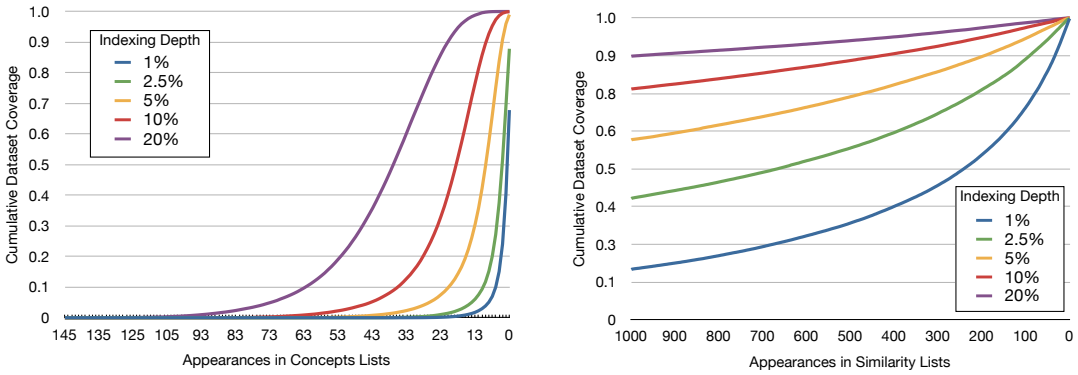


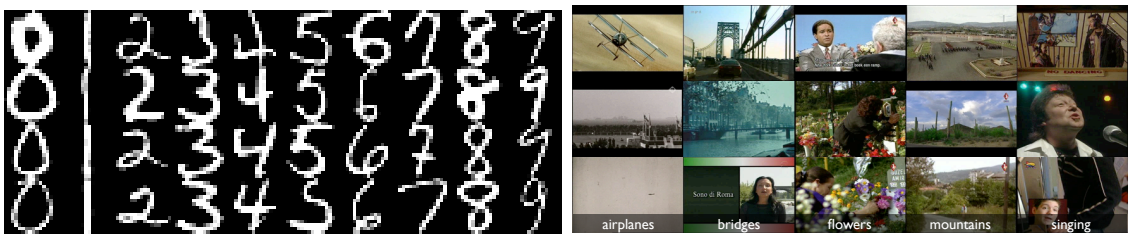
Figure 5.5: Subshot coverage and appearances in concept and low-level similarity result lists of the TRECVID2008 dataset at different indexing depths.

distinct behaviors from the concept and similarity indexes. For concepts, at the 5% indexing depth, about 90% of all subshots appear with repeats in at least five concepts and at the 10% indexing depth, over 90% of all subshots appear in at least ten concepts. Given that there are only a few hundred concepts available, these small numbers empirically demonstrate that the result lists for each concept are fairly diverse. If this were not the case, the concept coverage figure would slowly asymptote to one instead of exponentially climbing at small concept counts. Analyzing the same coverage task with low-level color similarity, the result lists are far less distinct and therefore experience a high number of repeat appearances before achieving full dataset coverage. At the 2.5% indexing depth, about 55% of all subshots appear with repeats in 500 or more result lists. For example, if any sample were chosen at random, there is a 55% chance that this sample would appear in the truncated similarity list of over 500 other samples. While 500 repeated entries seems wasteful, it is only 0.4% of the total TRECVID2008 dataset so these repeats are miniscule compared to the storage costs of the entire dataset. With this analysis, an explicit tradeoff has been verified: if indexing depth is reduced, the number of diverse concepts and feature

modalities must be increased to achieve adequate dataset coverage so that no subshots are unavailable during subsequent search sessions – both of which are satisfactory solutions for the real-time guidelines set forth in section 4.4.

5.3 Experiments

In this section, the proposed proposed algorithms NE-LGC and EGR and two state-of-the-art algorithms, LGC and PVM, are evaluated with a number of different conditions by varying graph related and exemplar related parameters. Experimental classification performance is reported on both a multi-class dataset (US Postal Service, USPS, handwritten digit recognition) and multiple single-class retrieval problem (concept classification on the TRECVID2008 development set [101]). Figure 5.6 demonstrates class samples from these two datasets. The USPS dataset consists of roughly 200 hand-written examples for each of



(a) (b)

Figure 5.6: Examples of datasets used in reranking experiments from (a) USPS handwritten digits and (b) TRECVID2008 development data for high-level feature classification.

ten digits with features coming from greyscale pixel values, for a total of 2007 image samples of 16×16 pixels concatenated into a single 256 dimensional vector. The TRECVID dataset consists of 43616 keyframes extracted from documentary and educational videos where each class is one of 20 exhaustively labeled semantic concepts like *bridge*, *airplane*, and *flower*. The features for each sample are a 500-dimensional BoW SIFT histogram derived from difference of Gaussian interest points, as computed for the CU-Vireo374 classifiers models [55]. Unless otherwise noted, performance for parameter tuning experiments is reported using the USPS dataset because of its well-known nature and a dataset size that is reasonable for working set analysis. The default parameter settings for different techniques are included

in table 5.2 and are modified only during tuning for a specific experiment. Finally, for each parameter setting, 200 sets of randomly selected labels (20 runs with 10 relevant and non-relevant labels per class in each run) were evaluated with each algorithm to eliminate the chance of over-fitting. Parameters for the PVM approach were found through an exhaustive search of the parameter space with the same validation process.

parameter	**setting**	**technique**	**description**
μ	0.9	LGC, NE-LGC, EGR	tradeoff parameter for smoothness and label influence
k	5	LGC, NE-LGC, EGR	number of neighboring samples for graph construction
m	500	NE-LGC, EGR, PVM	number of exemplars for use in exemplar-based technique
c_1, c_2, β	100, 0.1, 2^{-5}	PVM	custom-tuned cost parameters

Table 5.2: Default parameter settings and their description for experimental evaluations of state-of-the-art and proposed relevance feedback algorithms.

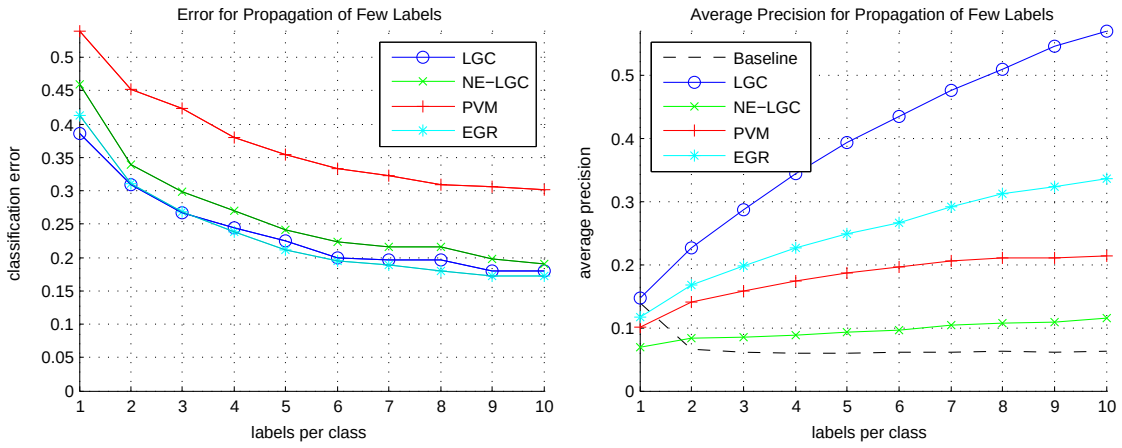
5.3.1 Overall Performance

This section provides performance numbers and execution times achieved under optimal conditions for the USPS and TRECVID datasets. These optimal conditions were found by exhaustive parameter searches with cross-validation as described above.

5.3.1.1 Quantitative Performance

While most experiments in this section were performed on the smaller USPS dataset, overall performance is also reported for the concept classification task on the TRECVID2008 development dataset in an effort to validate that the exemplar-based approaches are suitable for large, realistic datasets. It should be noted that the TRECVID experiments have classes that are not mutually exclusive, so mean average precision (introduced in 2.3) is utilized to report performance over the set of 20 concept classes. In each TRECVID experiment, working sets of 2000 samples (as described in 5.2.5.1) are derived using one randomly chosen

positive sample as queries for each class. This sample is used as a query over the entire dataset to get an "initial" result list so that the 2000-sample working set is non-random. This is an important difference from the USPS experiments, because this extra step requires different graphs to be constructed for each pair of class and experiment. Figure 5.7 illustrates the mean classification error over 10 USPS digits and mean average precision over 20 TRECVID2008 concepts. In this figure, the number of labels indicates the number of relevant and non-relevant labels included. For example, a setting of 5 indicates 10 total labels. In the TRECVID2008 experiments, relevance scores in the *baseline* approach use the equal fusion of content similarity (section 2.4) of all samples with relevant labels.



(a) (b)

Figure 5.7: Graph-based reranking performance demonstrating (a) mean classification error for the USPS multi-class dataset and (b) mean average precision of concept classification for the TRECVID2008 development dataset.

While both experiments show consistent performance gains as additional labels are included, the relative ranking of methods is different due to the evaluation metric. This paragraph examines multi-class classification of the USPS dataset. The relative ranking of different approaches in this figure also allude to ranking in future parameter variation experiments: the proposed EGR, classic LGC, proposed NE-LGC, and finally PVM. This observation is reassuring because it verifies that the proposed methods are comparable (NE-LGC) or better (EGR) than the classic LGC approach, with less than 5% absolute difference.

A second observation demonstrates that each approach generally shows improved performance with more labels. This steadily decreasing error rate validates that NE-LGC and EGR behave similarly when compared to classic LGC and that both approaches effectively propagate both small and large numbers of relevance labels.

Turning to performance in the TRECVID2008 experiments, an interesting effect is observed. Contrary to the strict sample-based classification error used in USPS experiments, the performance metric used here is based on ranked relevance scores across the dataset. While both experiments assign a relevance score to each sample in X, in the TRECVID2008 experiments relevance scores across the entire working set are used to rank results and provide a single average precision score. This metric demands that relevance scores are not only class-accurate but also globally consistent among a single working set. The effect of this metric is demonstrated by the dramatically different performances between different approaches as well as the bottom ranking of the NE-LGC approach. First, the EGR demonstrates a 42% relative shortfall when compared to classic LGC. Although EGR performance continues to climb with more labels, its AP is worse than the baseline CBIR approach for tests with fewer than 5 relevance labels. Although deferred for verification in future work, this specific finding may not be that unexpected knowing that the constructed affinity graph in EGR uses only 5 nearest neighbors to any one sample. Second, NE-LGC AP scores are consistently worse than the baseline AP scores. This observation demonstrates that propagated relevance scores for NE-LGC are highly localized and there is very little score normalization across all samples. Although some information loss was expected (all samples of X are now represented by a linear combination of a smaller $\tilde{X}$), this severe AP loss relative to the baseline method renders NE-LGC unacceptable for global result reranking. Briefly mentioned at the end of section 5.2.3, it is apparent that the mapping-in and mapping-out process has a detrimental "blurring" effect on propagated labels. In EGR, blurring naturally occurs because the labels from a single exemplar are distributed to a number of neighboring samples with the linear transform H. NE-LGC exacerbates this problem by first mapping in the labels from X into $\tilde{X}$ with a second transform in E.

While the results of these two experiments do not perfectly agree, performance based observations help to conclude that EGR is a viable substitution for classic LGC for all

reranking tasks and NE-LGC is viable for the case of strict classification tasks. As the introduction of this chapter indicates, these conclusions are reached by tempering performance with runtime considerations for use in a real-time environment.

5.3.1.2 Runtime

In this section, summaries on graph construction and evaluation time are reported. Each approach was implemented with no attempt to optimize code and evaluated with Matlab 2008b on a 2.4GHz Dual Core CPU. In table 5.3, the clustering, construction, evaluation, and total time required for each approach is reported in seconds. To avoid any artificial performance gains, over 20 test executions on the USPS dataset were averaged for the times below. This table demonstrates that compared to the traditional LGC, other approaches

approach	**clustering**	**construction**	**evaluation**	**total**
LGC	-	178.89	0.04	178.93
NE-LGC	11.90	4.98	0.62	17.5
EGR	11.90	40.96	2.21	55.07
PVM	11.90	43.71	1.92	57.53

Table 5.3: Average runtime in seconds for construction, clustering, and evaluation of two approaches from prior works and the approaches *proposed* in this chapter on the USPS dataset.

evaluated in this chapter can execute between three to seven times faster. This speed savings makes many of the approaches practical for real-time evaluations while trading off some portion of performance. All approaches besides LGC also require some time to select exemplars, which is identified by the "clustering" times. It should be noted that authors of the PVM approach claim that random selection of exemplars is also sufficient, although this scenario was not evaluated. Instead, to compare technique performances on equal footing, the cluster-selected exemplars from the NE-LGC and EGR approaches are also used for PVM. A final difference from LGC is that all other methods require a non-miniscule (although still small) amount of time to map samples and labels respectively into and out of the exemplar space. While this mapping process is accomplished with a matrix

multiplication, for large sample counts this cubic $O(n^3)$ or at best, a bounded $O(n^2 log\ n)$ time may become an important factor [92].

5.3.2 General Graph Variations

With any LGC based graph, the choices of appropriate neighborhood size and tradeoff parameter settings play a critical role in final algorithm performance. The following sections present a number of experiments that catalog the effects of different parameters used for each approach. These tuning experiments are presented to demonstrate that algorithms from other works have been adequately evaluated and the handful of required parameters are generally robust to setting variations. Also in this section is an experiment analysis of error when using unequal counts of labels from relevant and non-relevant classes.

5.3.2.1 Graph Neighborhood Size

Selecting the right neighborhood size is a non-trivial task in graph formulation. For accurate manifold representation in a graph, it is critical to express enough edges (similarities) between the right number of nodes (samples) such that a sample has sufficient influence from its neighbors, but only those neighbors that are within a reasonable distance. Typically, datasets are represented within graphs using every sample and its nearest three to five nearest neighbors. Those datasets with a weak underlying manifold structure must often increase this neighborhood size for acceptable performance. Figure 5.8 illustrates that different neighborhood sizes have little effect on the error rates of graph-based approaches evaluated. From this figure, the notable findings are that the performance of LGC and PVM methods suffer dramatically with only neighbor in graph construction. Not surprisingly, including too few neighbors can prevent a graph from accurately connecting sample neighbors. Similarly, the inclusion of too many neighbors can pollute the constructed graph by incorrectly creating edges between samples of different classes. The graphs in this figure also confirm that although the PVM approach uses a dense exemplar mapping, its underlying graph is still susceptible to the same pitfalls in LGC based approaches.

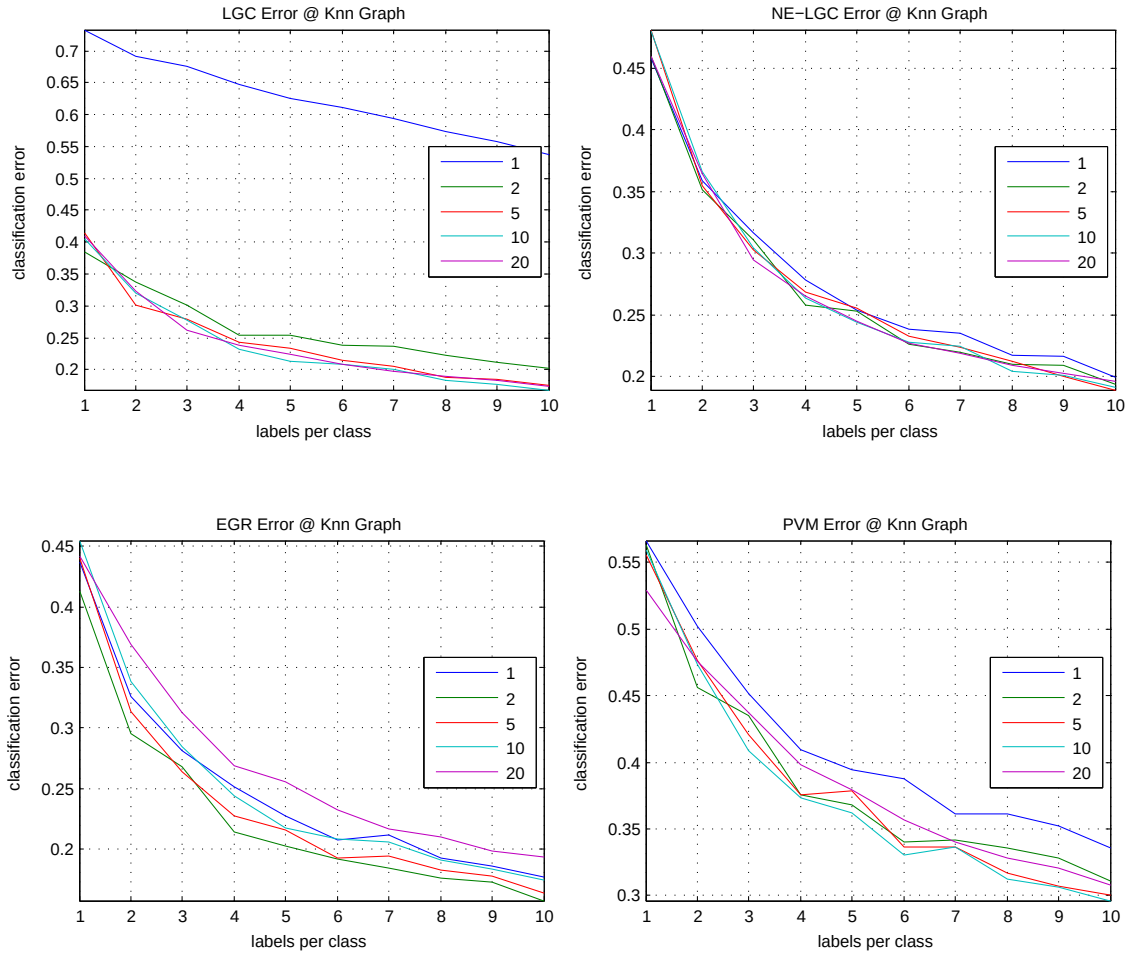


Figure 5.8: Classification error over existing and proposed graph-based methods when varying the number of neighbors during graph construction.

5.3.2.2 Tradeoff Parameter

The tradeoff parameter controls the influence of graph smoothness and the provided class labels and is inherent in all LGC formulations as shown in equation 5.2. In previous works, the tradeoff parameter μ is often chosen to highly favor provided sample labels. Figure 5.9 confirms that increased reliance on sample labels consistently yields better performance for all of the methods evaluated here.

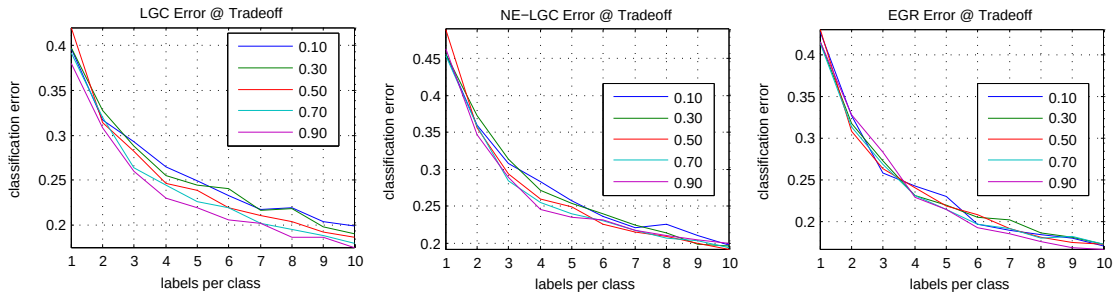


Figure 5.9: Classification error over tradeoff adjustments favoring manifold smoothness or label-based influence.

5.3.2.3 Unbalanced Classes

When searching datasets like internet images, user generated movies, and surveillance videos that are "in the wild", it is common for the number of relevant samples to be greatly outpaced by the number of non-relevant samples. For annotation and classification tasks, this disparity is generally referred to as unbalanced classes. Unbalanced class situations are detrimental to all machine learning approaches. Relevance feedback and reranking tasks can also suffer from unbalanced class scenarios and this effect on the proposed approaches is explored in figure 5.10 below. All parameters for these two USPS classification exper-

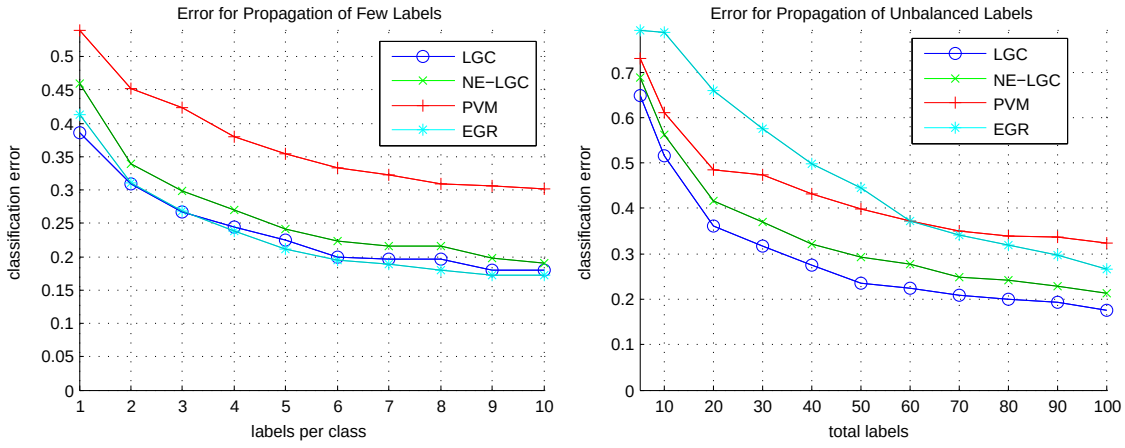


Figure 5.10: Classification error when the number of labels for each class are unequal.

iment sets are exactly the same. The only difference is that labels utilized in the second experiment set were chosen randomly from all classes, requiring the reporting of the num-

ber of total labels instead of the number of labels per class. In this figure, the relative performance of approaches in this figure does not perfectly agree with observed patterns in other experiments in this chapter. When an unbalanced number of labels are available, EGR performance is significantly lower than expected. Looking back at the formulation of EGR in equation 5.14, this behavior may not be so surprising. Inside of the inverse of G^*, the subset of labeled samples is used to influence the smoothness of the mapped affinity graph with $H_l^T H_l$. If an insufficient number of samples from one class are included, graph smoothness will be biased towards that class. This bias is removed as unbalanced classes are better filled out, as confirmed by the convergence of balanced and unbalanced EGR performance as label count increases.

5.3.3 Exemplar Specific Variations

Whereas the last section analyzed the performance gains by varying several graph parameters, this section analyzes the effect of parameters unique to exemplar-based approaches including exemplar count, exemplar mapping strategies, and a discussion about the effect of exemplar density. During this analysis, formulations from other works using exemplar mapping (PVM classification [138]) and feature quantization are also evaluated.

5.3.3.1 Mapping Techniques

The mapping technique proposed in section 5.2.2 is only one of many ways to construct projections between the original sample space and the exemplar space. Two main properties can be altered in the mapping process: the weighting scheme to account for neighbors in the space and normalization of the each sample's mapping vector. In this work, normalized ridge regression was chosen for performance robustness, after experimental validation in this section. Using equation 5.8 as a template, several mapping techniques listed in table 5.4 were evaluated for determining the mapping coefficients in H and E. For these equations, k is the number of nearest neighbors in the exemplar space, $\mathcal{N}$ is an ordered list of decreasing similarity. Normalization in these equations means row-normalization, such that each column is divided by the sum of its row $h_{ij} = h_{ij} / \sum_j h_{ij}$.

To provide an objective scoring of each technique, mapping matrices E and H were

computed as described in section 5.2.2.2 and average error and AP scores were calculated after applying these mappings to a known label set. In twenty tests, a random dataset was constructed with 2000 samples and 256 dimensions per sample. Binary relevant and non-relevant labels Y were randomly assigned to all samples in each set, where the size of each relevant class was only 10% (or 200 samples), which better simuluates the TRECVID and USPS datasets. Finally, for each random dataset, performance scores were averaged over five random label configurations. This process effectively evaluates the information loss from mapping in $Y_{in} = EY$ and mapping out $Y_{out} = HY_{in}$ with each technique. This

technique	mean error	mean AP	equation
regression	0.082	0.323	$h_{ij} = regress(x_{\mathcal{N}_j(i)}, \tilde{x}_j)$
normalized regression	0.082	0.378	
exponential weighting	0.098	0.313	$h_{ij} = \frac{1}{2^{i-1}} exp(-d(x_{\mathcal{N}_j(i)}, \tilde{x}_j)/\sigma^2)$
linear weighting	0.099	0.273	$h_{ij} = \frac{k-(i-1)}{k} exp(-d(x_{\mathcal{N}_j(i)}, \tilde{x}_j)/\sigma^2)$
equal weighting	0.100	0.217	$h_{ij} = exp(-d(x_{\mathcal{N}_j(i)}, \tilde{x}_j)/\sigma^2)$
normalized equal	0.100	0.236	

Table 5.4: All exemplar mapping techniques evaluated for performance. Average mapping fitness was evaluated with synthetic datasets using mean error and mean average precision.

table indicates that the regression-based mappings best preserve sample labels through both mapping stages, which is not surprising because regression finds the best fit for each mapping. It is interesting to note that even in such a simple mapping test, all techniques experience some loss of information both in terms of pure classification (mean error) and relative ranking among different results (mean AP). Unfortunately, these numbers also indicate that the performance of any graph-based exemplar approach is bounded by errors from this mapping process.

Comparing mapping techniques with experiments on the USPS dataset, the impact of each is more clearly seen in figure 5.11. In NE-LGC experiments, there is a clear error stratification between techniques, where performance improves between equal, linear, exponential, and finally regression techniques. Across all label counts, this ordering is roughly stable for NE-LGC. In EGR experiments, mapping choice has less of an effect on resulting

classification error. Here, error differences for the most labels varying less than 2.5% in absolute error. The relative order of these techniques in terms of decreasing error is also stable, with the exception of normalized regression.

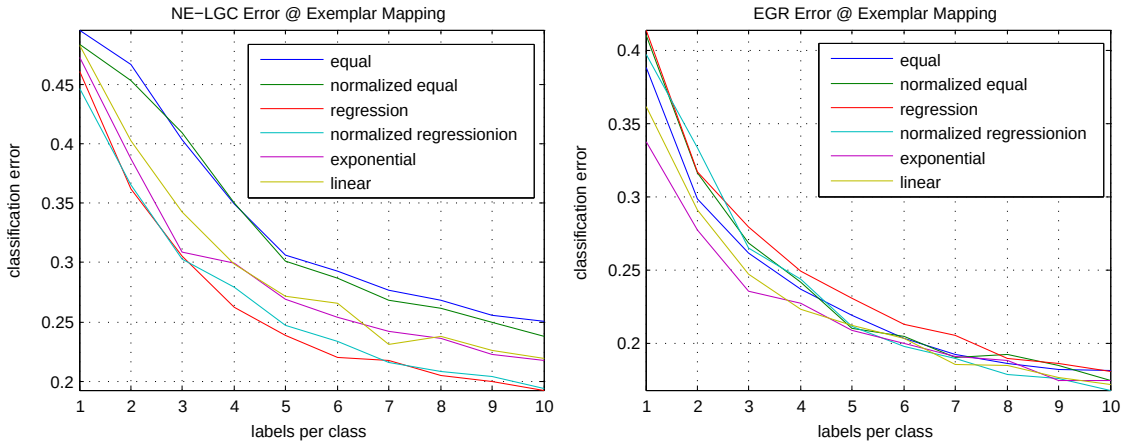


Figure 5.11: Classification error for exemplar mapping and normalization techniques.

5.3.3.2 Exemplar Count

Exemplar count refers to the dimension m or count of $\tilde{X}$ selected for mapping of the original sample space X. These selection and mapping processes are sample reduction techniques, so the same set of rules is applied to choosing the best exemplar count. With too few exemplars, the sample space in not sufficiently covered, potentially missing pockets of low-density samples. With too many exemplars, the savings in computation speed and resource requirements that motivated exemplar-based approaches are lost. Figure 5.12 demonstrates performance tradeoffs for larger exemplar counts using the USPS dataset of 2007 original samples. These graphs demonstrate that both NE-LGC and EGR benefit from larger numbers of exemplars. The error reduction from increased exemplar count begin around 15-20% when incrementing from 100 to 200 exemplars (a 100% increase), but this error reduction falls to less than 5% when increasing from 400 to 500 (only 25% increase). While experiments could have continued to contain an equal count of exemplars and samples, the corresponding construction and evaluation times reported in section 5.3.1 would no longer be suitable for real-time environments. Returning to the PVM graph in

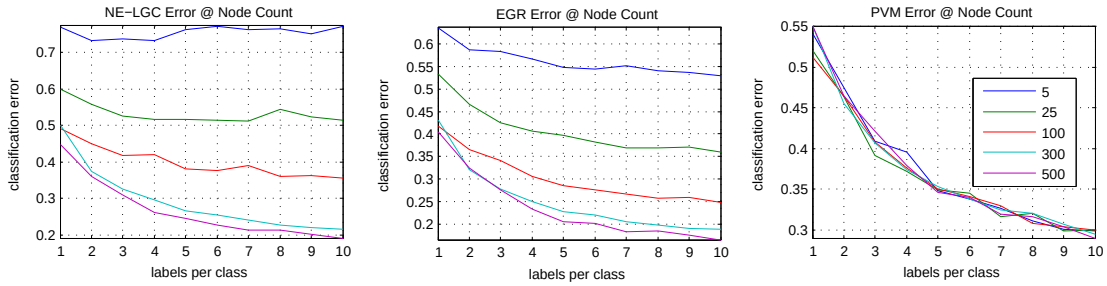


Figure 5.12: Classification error over various exemplar counts for different methods.

figure 5.12, performance indicators agree with those from PVM authors citing that increased exemplar count has few benefits. This figure also indicates that the density of the exemplar mapping H can play a large role in performance and continued discussion is provided in the next section.

5.3.3.3 Exemplar Density

While different mapping techniques can be applied, the most important factors in exemplar-based approaches are exemplar count and mapping density. The EGR and PVM approaches use different parameterizations and approximation techniques but at their core they are strongly related. Even with this strong similarity, the PVM approach exhibited weaker performance than EGR throughout experiments in this section. The root of this problem stems from the number of exemplars influencing the label of a sample in the original space. While prior works (including the PVM approach) use exemplars to approximate original samples in the *feature space*, these experiments have shown that it is not appropriate to use all exemplars to approximate samples in the *label space*. Figure 5.2 demonstrates this essential aspect that allows NE-LGC and EGR approaches to outperform PVM. Looking at this problem more abstractly, this problem can be likened to the semantic gap between representation and understanding. Just as semantic concepts were introduced in section 2.1.3 to combat shortcomings with low-level feature indexing, a sparse exemplar density is required to overcome differences in semantics that are not fully captured by graph-based manifold techniques.

5.4 Related Work

This chapter proposes near real-time solutions for utilizing relevance feedback and reranking that use sample exemplars for efficient dataset scaling and high-quality label propagation in graph-based formulations. A review for several dataset scaling and feature reduction techniques was included in section 5.2.2.1 so the remainder of this section is devoted to additional works that focus on high-quality relevance propagation among exemplars with only a handful of relevance labels.

Central to the approaches in this chapter is the efficient use of a graph. Few approaches have sought to optimize construction and evaluation time, as done in this chapter, and have instead focused on refining graph structure to correct missteps in user labels and noisy feature data. First examined in section 5.3.2.3, it is commonplace for classes to have unbalanced label counts. One work for the classification of cell phenotypes utilizes incremental updates of F^* by dynamically re-weighting the influence of user labels Y according to a sample's degree (or density) in the graph [121]. While this approach can help to balance class influence, the use of a node's degree within the graph could introduce dangerous artifacts if the attributed labels are noisy or both relevant and non-relevant samples are very proximal in a dense sample neighborhood. Subsequent work by these authors proposed a self-diagnosis procedure that dynamically switched operational modes between label correction and normal label propagation [122]. In the initial label correction stage, label propagation is performed, but instead of returning new relevance scores, the method would correct labels that strongly disagreed with the resulting relevance. After some stability criterion is met, the method would return begin normal label propagation but with a much cleaner label set.

Complementary to work analyzing label quality in a graph are related works that focus on graph properties themselves. A recent work attempts to guarantee very a high-quality graph through incorporation of a few rules during construction [71]. First, instead of step 3 in section 3.3.2.3 which makes creates a symmetric weight matrix, the authors enforce the following symmetry constraints utilizing the nearest neighbor lists $\mathcal{N}_i$ and $\mathcal{N}_j$ for sample x_i

and x_j respectively.

$$W_{ij} = \begin{cases} A_{ij} + A_{ji}, & \text{if } j \in \mathcal{N}_i \text{ and } i \in \mathcal{N}_j \\ A_{ji}, & \text{if } j \notin \mathcal{N}_i \text{ and } i \in \mathcal{N}_j \\ A_{ij}, & otherwise \end{cases}$$

This alternate formulation for W strongly rewards those edges between nodes (samples) that have each other in their respective k nearest neighbor lists. A third rule proposed enforce a unit-normalization of of the weight matrix such that $W1 = 1$ to ensure that this matrix is doubly stochastic. With a doubly-stochastic matrix, the problems of unbalanced classes labels and incorrect influence of samples from different classes in the same neighborhood are significantly mitigated. While all of these solutions could be integrated in future revisions of NE-LGC and EGR, special care must be given to verify that the above graph optimizations still apply in the exemplar space.

Finally, the current performance gap between classic LGC and the proposed NE-LGC and EGR methods can most likely be aided by additional study of deconvolution techniques to refine the mapping process. This bridge to other signal processing literature is necessary because it is unlikely that direct mapping techniques alone, like the proposed sparse exemplar mapping or dense mapping PVM method, will sufficiently resolve disparities between exemplar and traditional performances. One promising work in this area involves the inclusion of multiple sources to better deconvolve the original signal [86]. In application to this work, multiple sources could be multiple exemplar sets and the signal to deconvolve is the relevance labels over the various exemplar sets. A second work uses a set of examples available during training for reconstruction of the original signal [35]. While this technique may only directly apply to pre-defined classes like semantic concepts or handwritten digits and not general search targets, the benefits of its supervised nature may be able to outweigh problems introduced by dynamically selecting exemplars.

5.5 Summary

In this chapter, solutions are offered that reduce traditional problems of irrelevant result triage and the wasted time during search target teaching sessions. Using automatic relevance

feedback instead of dramatic query refinement or expansion, only a few relevance labels are required from the user to achieve high-quality label propagation over a large set of samples. The techniques proposed in this chapter differ from traditional ones because they have been simultaneously optimized to accommodate both large scale problems (either in the sample space or feature space) and low-latency requirements of real-time environments with the use of strategically chosen content exemplars directly from the searched dataset. By choosing exemplars from the dataset, the proposed graph-based approaches use a semi-supervised analysis to exploit relationships between both labeled and unlabeled data. Benchmarking against existing state-of-the-art work in graph-based LGC, the proposed methods offer hybrids requiring one sixth of processing time (NE-LGC) and realizing 101% of classification performance (EGR) with non-optimized implementations. Both of the proposed approaches were benchmarked on a well-studied multi-class digit recognition problem and a real-world semantic concept classification problem, demonstrating robustness in both semi-supervised environments.

This chapter presented two real-time approaches as suitable candidates for integration in to the CuZero prototype system discussed in the last chapter. This combination is both feasible and complementary. Thanks to the minimal processing time required to quickly rerank any one search target such approaches can be introduced with little additional latency to the user. Also, due to the graph-based label propagation along a sample's manifolds, initial errors from retrieval methods using low-level features or non-customized semantic concept classifier scores can be corrected. Future work could also involve the revision of the proposed exemplar methods to account for additional robustness in the exemplar mapping process, graph construction [71], or label correction [122]. The latter problem will almost always occur in search systems because labeling errors may be introduced by users that are trying to quickly evaluate a new search topic. Poor performance often occurs if these errors are not corrected or identified with automated processing.

Chapter 6

Conclusions and Future Work

In this thesis, three major proposals that were discussed: guided query formulation, real-time query space exploration, and reranking results with label propagation. This chapter summarizes the main ideas, discusses contributions and then concludes with future directions of research for these topics.

6.1 Summary of Thesis

This thesis investigated a number of problems associated with the efficient and engaging execution of a multi-level interactive video search system. Several of these fundamental problems were addressed and their solutions are reiterated here. When possible, these solutions were also integrated into a prototype system called CuZero, which was evaluated both formally by an expert and novice user and informally with a number of challenging real-world data sets.

6.1.1 Guided Query Formulation for Informed Users

To help users discover the most relevant keywords and semantic concepts to answer a search topic, guided query formulation was proposed. Input from the user is mapped into a set of mid-level semantic classifiers that are automatically suggested by the system. This mapping process not only uses lexical matching techniques but also discovers related concepts using statistical co-occurrence and data mining techniques. The suggestion process offers

the system's best guess at related concepts but does not force any of these suggestions into the user's query. This helpful and informative approach facilitates a truly interactive and guided query formulation. Leveraging the user's ability to quickly see and interpret visual disparities, suggested concepts are visually scaled according to performance and prior frequency on training data and frequency. As a fall-back suggestion method, the user's input is also evaluated and expanded with a text-only search to discover related keywords in the target dataset. All of these methods help novice and expert users alike to better understand the content of the target dataset, in terms of relevance to the user's current input and expected performance with a set of automatically suggested semantic concepts. Comparing these automatic concept suggestion offerings to state-of-the-art solutions, this thesis presents the most comprehensive tools for mapping including lexical, statistical, and on-line data-mining techniques.

6.1.2 Intuitive Query Space Exploration

Available during both query formulation and result exploration, an intuitive visualization called the query navigation map was introduced. In existing result inspection interactions, a user may not understand how the returned results relate to his or her selected query criteria. This confusion is one problem that can lead to a potentially frustrating blind inspection of results. With the proposed query navigation panel, all utilized query criteria (query anchors) are visualized on a 2D grid and users have a direct understanding of what query anchor contributed to the currently inspected set of results. Further, this query space can also be manipulated by the user to construct his or her query. This formulation allows full control of the number of anchors and their respective proportion of influence, which is ultimately evaluated to produce relevance scores for result browsing.

6.1.3 Simultaneous Exploration of Many Query Permutations

The visualized query space introduced in this thesis also helps the user to avoid the process of trial-and-error search. Navigation in the query space gives immediate feedback for different combinations of the different query criterion so the user always knows what part of the query is helpful or hurtful. The visualization utilizes an intuitive 2D display, so even novice users

can easily grasp the only usage instruction "closer to a query anchor is more like it". Using a non-linear transformation of 2D Euclidean distances, the system can easily compute the influence of each anchor and derive the correct relevance scores for results from the target dataset. The query navigation map also offers a straight-forward solution to the fusion of results from multiple query modalities. For example, search results from text, concept, low-level feature, and other query modalities can easily be combined and displayed with no modification of the navigation panel's algorithms. Evaluations of CuZero on a number of difficult datasets discover beneficial query configurations that were previously not available with conventional search systems.

6.1.4 Real-time, Automatic Reranking with Relevance Feedback

Finally, even with these innovations in query formulation and query space navigation, search results available to a user may not perfectly align with his or her search target. Thus, a new graph-based approach using efficiently selected exemplars was introduced to personalize search results according to small numbers of user relevance labels. First, in modern literature graph-based techniques like LGC (local and global consistency [139]) have been shown to achieve very high performance with only a few user relevance labels because they leverage the similarity of results along an underlying sample manifold. Unfortunately, the construction of most graph-based techniques is plagued by matrix operations with high computation and resource requirements. Combatting this problem, original sample sets are approximated with a small-scale set of exemplars and a sparse mapping between these two sets is computed. Two unique formulations optimized for speed and performance were presented and their performance was evaluated on a realistic concept classification dataset (TRECVID2008 development) and a well-known hand-written digit dataset (USPS digits) and contrasted against current state-of-the-art methods. In USPS evaluations, the proposed methods were able to execute in one-sixth to one-third of the required time and achieve respective performances from 96% to 101% of the state-of-the-art. In a search-based TRECVID reranking task, the performance of the graph-based exemplar approaches were less than those seen on the USPS dataset, but still showed gains over a nearest neighbor baseline. Findings indicate that these approaches are viable for real-time environments that

where very few relevance labels are available, making them ideal for integration into future versions of the CuZero prototype system.

6.1.5 Major Contributions

The major contributions summarized from this thesis can be attributed to a few critical topics in interactive search.

- **Guided, informed query formulation** defined a comprehensive system that eases both system and human efforts for query formulation. This natural symbiosis familiarizes users with the system's capabilities and search data and allows the system to quickly get relevant query parameters from the user.
- **Instant inspection of multiple queries** removed the burden of parameter tuning from the user and provide an intuitive visualization for diverse query exploration. With this approach, users spend more time inspecting potentially relevant results and little or not time figuring *how* to explicitly find them.
- **Automatic reranking with relevance feedback** delivered a personalized result ranking without breaking the user's thought flow and concentration on the search topic. The proposed state-of-the-art label propagation approaches can quickly rerank large datasets with only a handful labeled samples.

6.2 Future Directions

In future work, additional refinements to CuZero could be integrated to further enhance either query formulation, result exploration, or relevance feedback. The query formulation stage was designed in such a way that it can easily accommodate other suggestion techniques without modifying the underlying algorithms. Similarly, modular algorithms in result browsing allow fast evaluation of alternate techniques for planing influence of query anchors or visualizing results with additional consideration to result cognition by users (i.e. altering spatial layout to facilitate faster inspection times). Finally, the proposed relevance feedback technique can be further enhanced by using rearnking to bootstrap subsequent

machine learning methods to further improve predicted relevance scores with no additional cost to the user. The remainder of this section details these revisions to serve as practical examples of easy refinement steps within the modular CuZero framework.

6.2.1 Query Formulation

The use of a textual mapping process can be problematic due to the natural evolution of vocabulary. Similar to the "flickr distance" discussed in section 4.6, a new technique has proven to be an effective way to harvest a large lexicon for mapping and simultaneously apply that mapping in a sensible way [53]. This approach simultaneously bridges unknown vocabulary from an external source while tempering that vocabulary for the user's active dataset. Additionaly, the suggestion methods in section 4.2 cover a range of statistical and lexical techniques, but an organically grown source of content suggestions can come be derived from long-term user experiences. For example, if a user is acting as an analyst or librarian, it might be useful to let the user construct their own library of formulated query navigation maps that can easily be used in query formulation for a new search topic – even if it is only tangentially related. One particularly powerful argument for allowing a navigation map library is the idea of a *superanchor*. A superanchor is an iconic representation of an entire query navigation map: it captures the defining query anchors of a query, the weights of those anchors, and their spatial configuration. Superachors could then be easily be reused in subsequent query formulations explicitly recalled (by the user) or as a suggested query anchor itself, emulating an additional semantic classifier, as illustrated in figure 6.1. Using the concept of a superanchor, the CuZero prototype could also quickly bootstrap *personalized media filters* for unseen data that have not been indexed into the archive. In addition to the configuration of a user's query (and query anchors), a users's labels over seen data could be leveraged to construct a fusion strategy and score threshold so that unseen data could be analyzed with existing tools and quickly filtered to reduce the quantity of data a user must inspect. A final, more passive way to discover new query suggestions, is often referred to as *collaborative filtering*, which uses both long-term relevance learning and similarity across prior queries [56]. Confirmed by empirical studies of online Web document queries [109], [29], example features for collaborative filtering experiments can come from

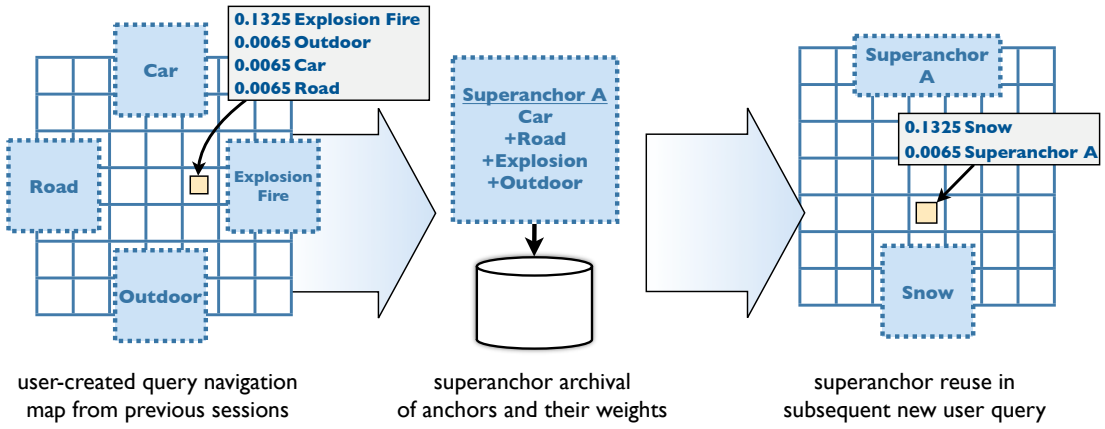


Figure 6.1: Archival and modular reuse of a query navigation map as a superanchor.

formulated queries (71% of previously selected documents had the same query keywords) or result selections (for the same user, 35-40% of documents selected were selected before; for different users, 7% repeated selections). In a non-passive scenario, collaborative filtering could also be accomplished with either the superanchor or personalized media filter methods above, which could be passed between users and iteratively refined.

6.2.2 Result Visualization & Exploration

In addition to suggestions during query formulation, query exploration techniques in chapter 4 may also be further enhanced. First, for the organization of the query navigation map itself, methods may be available that produce optimizations for relevance planning (section 4.3.1.2). Dimensionality reduction and projection techniques, such as those from section 4.6, may help to better plan the relevance scores and cell assignment of results after the user has formulated his or her query navigation map. Currently, a weighted fusion approach is used to determine the relevance score for a result in each navigation cell. However, alternative planning strategies may be investigated to provide alternative solutions for relevance within a cell, like the repeat minimization discussion in section 4.3.1.3, Along these lines, the query navigation map could be augmented to allow multiple "zoom" levels organized by a hierarchy that allows the user to get a faster breadth-based exploration of the query space and its results. Second, although the query navigation map allows user to explore results via anchor

relationships, but not relationships between the results themselves. It might be beneficial to further organize results in the result browsing window by optimally choosing small image regions to display in browsing [70] or grouping image results into semantically related visual islands [135]. Third, the current query navigation map has the limitations that anchor positions are static and users can not compose high-order (non-linear) permutations. For the former, users can not select cells with high weights on two distant anchors. Currently, to accommodate this change, the user must reformulate the query and reposition one of the two anchors. For the latter problem, users can not easily manipulate anchors to achieve non-linear weights on more than two anchors. Although it is unlikely that non-expert users would need this functionality, it could be accomplished by customizing the σ value for each anchor in algorithm 4.1 such that the influence of an anchor had decayed more rapidly or more slowly. One possible corresponding visual change would be to physically shrink or grow the anchor's icon in the navigation map. Finally, embracing the interactivity of the query navigation map, the process of multiple users working together to answer a single search topic (*collaborative search*) could be evaluated. Recent work in live search scenarios has shown that the inclusion of multiple users in a search task working on different cognitive processes (i.e. formulating, exploring, inspecting) can achieve high performance [99], [89]. Collaborative multimedia search, although a topic in its infancy now, will likely continue to grow in impact with the immersion of multimedia and ubiquitous nature of mobile devices. The increasingly social nature of events and their digital image and video representation has already demonstrated itself with the popularity of online social site like Facebook and twitter, image sharing sites like flickr, Picasa, and Hurly, and even news reporting sites like CNN iReport. Although this brief list demonstrates that numerous opportunities for collaborative image applications are available, no single service is currently large or stable enough to become mandatory on all new mobile devices.

6.2.3 Efficient Reranking With Diverse Features

With the ability to quickly construct and evaluate graphs from either small working sets or over the entire dataset in chapter 5, these graphs can now be adapted to combine different cues from multiple feature modalities. The core problem with feature combination is where

to fuse information. Traditional approaches either directly fuse all features by creating a single concatenated vector or apply a score weighting scheme after computing a relevance score for each modality (section 2.4). While a weighting approach traditionally yields higher performance, the core problem for an application of the proposed exemplar graphs is that predictions from each modality's graph are not normalized between zero and one (unequal relevance scores). One solution might be to reformulate the exemplar-based techniques using a GFHF framework so that all propagation results are between zero and one and thus a weighting fusion will not be biased. Another solution, utilizing the existing LGC framework and integrating multiple modalities is to concatenate and zero-pad different sets exemplars themselves so that a single manifold is learned over multiple feature modalities. For example, if there are three feature modalities to be considered, A, B, C, exemplars for each modality can fist be selected $\tilde{X}^a \in \mathbb{R}^{m \times d_a}, \tilde{X}^b \in \mathbb{R}^{m \times d_b}, \tilde{X}^c \in \mathbb{R}^{m \times d_c}$. Next, all exemplars sets are merged into a set of meta-exemplars $\tilde{X}'$ where the exemplars from each modality are expanded to match total dimensionality d and missing dimensions are zero-padded as below. Here, zero-padding each row set guarantees that affinity computations for graph construction are not effected by exemplars from different sets, essentially creating multiple set-based sub-graphs.

$$\tilde{X}' = \begin{bmatrix} \tilde{x}^a & 0 & 0 \\ 0 & \tilde{x}^b & 0 \\ 0 & 0 & \tilde{x}^c \end{bmatrix}$$

With this formulation, original samples can be directly concatenated to include all feature modalities $X' \in \mathbb{R}^{n \times d}$. Now, during label propagation, the influence of each modality can be weighted by manipulating the mapping matrix $H' \in \mathbb{R}^{n \times 3m}$ (and E) accordingly. This formulation is a straight-forward fusion technique for the proposed exemplar-based graph approach that fuses labels within a graph in a disciplined fashion.

6.2.4 Exemplar Graphs for Rapid Classifier Development

With the ability to quickly construct and evaluate graphs from either small working sets or over the entire dataset in chapter 5, coarse, fast concept classifications are now feasible for novel data. One practical application example is the speedy creation of a new concept

classifier. Traditionally, the creation of a new classifier (as described in section 2.1.3) requires a large number of training samples and a potentially long training process. After creation, the application of a large ontology of classifiers (such as the *Columbia374*) can also be computationally expensive, depending on the complexity of each classifier. The exemplar-based graphs presented in this thesis provide initial solutions for both of these problems. To bootstrap a new classifier, the proposed exemplar-based method needs only a handful of labeled samples. Since the construction of an exemplar graph depends on an adequately sampled feature space and not the number of labels, several exemplar-based graphs can be constructed and quickly evaluated with user-labels for a previously unknown semantic concept. If satisfactory, the output of this process can then be used as input for subsequent model training, but with the addition of the propagated labels. A second novel application of exemplar-based graphs is to quickly propagate a large number of classifier labels to new data. In this usage, the exemplar graph could be constructed with a small number of samples with labels from each of the target classifiers. For example, if targeting 100 classifiers, five relevant samples for each class could be collected and directly used as exemplars. Unlike the reranking scenarios in the previous chapter, these graphs will be explicitly constructed with exemplars from different classes, all of which already have labels. Next, as new samples without classifier labels are provided, the exemplar-based propagation techniques proposed here are used to propagate labels for each of the 100 classifiers. In this case, a large number of classifiers has been compactly represented by exemplars and relevance high-quality labels are propagated to new data through the graph in a matter of seconds. Because these exemplars come from a fully labeled dataset, the quality of each sample can be estimated by its graph affinity to other similarly labeled nodes or through statistical and mutual information-based tests. Although there are parameter tuning options to consider, both of the formulations in this paragraph present new opportunities for large-scale, high-quality label propagation for both unseen semantic concepts and those with a pre-defined set of labels.

Bibliography

[1] John Adcock, Matthew Cooper, John Doherty, Jonathan Foote, Andreas Girgensohn, and Lynn Wilcox. Managing digital memories with the fxpal photo application. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 598–599, New York, NY, USA, 2003. ACM. 37

[2] John Adcock, Matthew Cooper, and Jeremy Pickens. Experiments in interactive video search by addition and subtraction. In *CIVR '08: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 465–474, New York, NY, USA, 2008. 47

[3] Aditya Vailaya Anil, Anil Jain, and Hong Jiang Zhang. On image classification: City images vs. landscapes. *Pattern Recognition*, 31:1921–1935, 1998. 15

[4] Stéphane Ayache and Georges Quénot. Evaluation of active learning strategies for video indexing. *Signal Processing: Image Communication*, 22(7-8):692–704, 2007. 39, 40, 93, 99

[5] Mihai Bādoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 250–257, New York, NY, USA, 2002. ACM. 104

[6] Simon Baker and Takeo Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002. 110

[7] Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. Automatic combination of multiple ranked retrieval systems. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 173–181, New York, NY, USA, 1994. Springer-Verlag New York, Inc. 27

[8] Paul Bausch and Jim Bumgardner. *Flickr Hacks*. O'Reilly Press, 2006. 54

[9] Nicholas J. Belkin, Paul Kantor, Edward A. Fox, and John A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, 31(3):431–448, 1995. 27

[10] Robert M. Bell and Yehuda Koren. Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.*, 9(2):75–79, 2007. 4

[11] Ana B. Benitez, Di Zhong, Shih-Fu Chang, and John Smith. Mpeg-7 mds content description tools and applications. In *CAIP '01: Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns*, pages 41–52, London, UK, 2001. Springer-Verlag. 37

[12] Kristin P. Bennett and Ayhan Demiriz. Semi-supervised support vector machines. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 368–374, Cambridge, MA, USA, 1999. MIT Press. 42, 98

[13] Deng Cai, Xiaofei He, and Jiawei Han. Semi-Supervised Regression using Spectral Techniques. Technical Report UIUCDCS-R-2006-2749, University of Illinois at Urbana-Champaign, July 2006. 107, 108

[14] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualizer, an information workspace. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 181–186, New York, NY, USA, 1991. 70

[15] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 20, 98

[16] Shih-Fu Chang, Lyndon Kennedy, and Eric Zavesky. Columbia university's semantic video search engine. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 643–643, New York, NY, USA, 2007. 45, 54

[17] Jau-Yuen Chen, C.A. Bouman, and J.C. Dalton. Hierarchical browsing and search of large image databases. *Image Processing, IEEE Transactions on*, 9(3):442–455, Mar 2000. 92

[18] Michael G. Christel. Examining user interactions with video retrieval systems. In Alan Hanjalic, Raimondo Schettini, and Nicu Sebe, editors, *Society of Photographic Instrumentation Engineers*, volume 6506, pages 650–606, 2007. 47

[19] Michael G. Christel, Chang Huang, and Neema Moraveji. Exploiting multiple modalities for interactive video retrieval. In *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pages 1032–1035, 2004. 59

[20] T. S. Chua, S. Y. Neo, K. Y. Li, G. Wang, R. Shi, M. Zhao, and H. Xu. Trecvid 2004 search and feature extraction task by nus pris. In *TREC Video Retrieval Evaluation Online Proceedings*, 2004. 33, 52, 90

[21] Ondřej Chum, James Philbin, Michael Isard, and Andrew Zisserman. Scalable near identical image and shot detection. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 549–556, New York, NY, USA, 2007. ACM. 3, 104

[22] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 20, 42, 98

[23] Madirakshi Das and Alexander C. Loui. Method for generating customized photo album pages and prints based on people and gender profiles. U.S. Patent 7362919, April 22, 2008. 2

[24] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, 40(2):1–60, 2008. 11, 17

[25] Ork de Rooij, Cees G. M. Snoek, and Marcel Worring. Mediamill: Fast and effective video search using the ForkBrowser. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 561–561, Niagara Falls, Canada, July 2008. 92

[26] Alberto Del Bimbo and Pietro Pala. Visual image retrieval by elastic matching of user sketches. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):121–132, 1997. 34

[27] Jia Deng, Wei Dong, Richard Socher, Li-jia Li, Kai Li, and Fei Fei Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255, 2009. 18

[28] Abdigani Diriye, Ann Blandford, and Anastasios Tombros. A polyrepresentational approach to interactive query expansion. In *JCDL '09: Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 217–220, New York, NY, USA, 2009. ACM. 53

[29] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 581–590, New York, NY, USA, 2007. 136

[30] Ran El-Yaniv, Dmitry Pechyony, and Vladimir Vapnik. Large margin vs. large volume in transductive learning. *Mach. Learn.*, 72(3):173–188, 2008. 104

[31] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. 18, 55

[32] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995. 34

[33] Geoffrey A. Fowler. Amazon acquires software maker snaptell. *The Wall Street Journal*, page B5, June 18, 2009. 35

[34] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):214–225, 2004. 104

[35] William T. Freeman, Thouis R. Jones, and Egon C Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002. 110, 130

[36] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag. 2

[37] David Frohlich, Allan Kuchinsky, Celine Pering, Abbe Don, and Steven Ariss. Requirements for photoware. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 166–175, New York, NY, USA, 2002. ACM. 3

[38] G. Fung and O. Mangasarian. Semi-supervised support vector machines for unlabeled data classification, 2001. 42, 98

[39] J.M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, and H. Geerts. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350, 2001. 15

[40] David Gibbon. Generating hypermedia documents from transcriptions of television programs using parallel text alignment. In *Research Issues In Data Engineering, 1998. 'Continuous-Media Databases and Applications'. Proceedings., Eighth International Workshop on*, pages 26–33, Feb 1998. 14

[41] Zhiwei Guan and Edward Cutrell. An eye tracking study of the effect of target rank on web search. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 417–420, New York, NY, USA, 2007. 41, 48, 63

[42] Sariel Har-Peled, Dan Roth, and Dav Zimak. Maximum margin coresets for active and noise tolerant learning. In *IJCAI*, pages 836–841, 2007. 104

[43] Donna Harman. Towards interactive query expansion. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 321–331, New York, NY, USA, 1988. ACM. 33

[44] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. *The elements of statistical learning: data mining, inference and prediction*. Springer-Verlag New York, LLC, July 2001. 107

[45] Alexander Hauptmann. Lessons for the future from a decade of informedia video analysis research. In *CIVR '05: Proceedings of the 4th ACM international conference on Image and video retrieval*, pages 1–10, 2005. 17

[46] Alexander Hauptmann, Wei-Hao Lin, Rong Yan, Jun Yang, and Ming-Yu Chen. Extreme video retrieval: joint maximization of human and computer performance. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 385–394, New York, NY, USA, 2006. 3, 65

[47] Alexander Hauptmann, Rong Yan, and Wei-Hao Lin. How many high-level concepts will fill the semantic gap in news video retrieval? In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 627–634, New York, NY, USA, 2007. ACM. 94

[48] Jingrui He, Mingjing Li, Hong-Jiang Zhang, Hanghang Tong, and Changshui Zhang. Manifold-ranking based image retrieval. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 9–16, New York, NY, USA, 2004. ACM. 44

[49] Matthias Hein and Markus Maier. Manifold denoising. *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, pages 561–568, September 2007. 107

[50] Atsushi Hiroike, Yoshinori Musha, Akihiro Sugimoto, and Yasuhide Mori. Visualization of information spaces to retrieve and browse image data. In *VISUAL '99: Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 155–162, London, UK, 1999. Springer-Verlag. 92

[51] Winston Hsu and Shih-Fu Chang. Generative, discriminative, and ensemble learning on multi-modal perceptual fusion toward news video story segmentation. In *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, volume 2, pages 1091–1094 Vol.2, June 2004. 14

[52] Winston Hsu, Lyndon Kennedy, and Shih-Fu Chang. Video search reranking via information bottleneck principle. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 35–44, New York, NY, USA, 2006. ACM. 43, 98

[53] Yu-Gang Jiang, Chong-Wah Ngo, and Shih-Fu Chang. Semantic context transfer across heterogeneous sources for domain adaptive video search. In *MULTIMEDIA '09: Proceedings of the 17th international conference on Multimedia*, October 2009. 136

[54] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 494–501, New York, NY, USA, 2007. 15, 20, 107

[55] Yu-Gang Jiang, Akira Yanagawa, Shih-Fu Chang, and Chong-Wah Ngo. Cu-vireo374: Fusing columbia 374 and vireo374 for large scale semantic concept detection. Technical Report 223-2008-1, Columbia University, August 2008. 117

[56] Diane Kelly, Karl Gyllstrom, and Earl W. Bailey. A comparison of query and term suggestion features for interactive searching. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 371–378, New York, NY, USA, 2009. ACM. 53, 136

[57] Lyndon Kennedy. LSCOM Lexicon Definitions and Annotations Version 1.0, DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia. Technical Report 217-2006-3, Columbia University, March 2006. 18, 19, 33, 45, 55

[58] Lyndon Kennedy and Shih-Fu Chang. A reranking approach for context-based concept fusion in video indexing and retrieval. In *CIVR '07: Proceedings of the 6th ACM*

international conference on Image and video retrieval, pages 333–340, New York, NY, USA, 2007. 44, 58

[59] Lyndon Kennedy and Mor Naaman. Less talk, more rock: automated organization of community-contributed collections of concert videos. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 311–320, New York, NY, USA, 2009. ACM. 3

[60] Lyndon Kennedy, Apostol Natsev, and Shih-Fu Chang. Automatic discovery of query-class-dependent models for multimodal search. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 882–891, New York, NY, USA, 2005. 33, 52

[61] Allan Kuchinsky, Celine Pering, Michael L. Creech, Dennis Freeze, Bill Serra, and Jacek Gwizdka. Fotofile: a consumer multimedia organization and retrieval system. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 496–503, New York, NY, USA, 1999. ACM. 37

[62] Neeraj Kumar, Li Zhang, and Shree Nayar. What is a good nearest neighbors algorithm for finding similar patches in images? In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 364–378, Berlin, Heidelberg, 2008. Springer-Verlag. 104

[63] Piyush Kumar, Joseph S. B. Mitchell, E. Alper Yildirim, and E. Alper Yldrm. Computing core-sets and approximate smallest enclosing hyperspheres in high dimensions. In *ALENEX), Lecture Notes Comput. Sci*, pages 45–55, 2003. 100, 104

[64] Jorma Laaksonen, Markus Koskela, Sami Laakso, and Erkki Oja. Picsom—content-based image retrieval with self-organizing maps. *Pattern Recogn. Lett.*, 21(13-14):1199–1207, 2000. 36, 92, 104

[65] Layar. http://layar.com/, September 22, 2009. Accessed: September 23, 2009. 35, 37

[66] Hyowon Lee, Alan Smeaton, Noel E. O'Connor, and Barry Smyth. User evaluation of físchlár-news: An automatic broadcast news delivery system. *ACM Trans. Inf. Syst.*, 24(2):145–189, 2006. 3, 37

[67] Jia Li and James Z. Wang. Real-time computerized annotation of pictures. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 911–920, New York, NY, USA, 2006. ACM. 18

[68] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1972–1979, 2009. 18

[69] Hao Liu, Xing Xie, Xiaoou Tang, Zhi-Wei Li, and Wei-Ying Ma. Effective browsing of web image search results. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 84–90, New York, NY, USA, 2004. ACM. 36

[70] Hao Liu, Xing Xie, Xiaoou Tang, Zhi-Wei Li, and Wei-Ying Ma. Effective browsing of web image search results. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 84–90, New York, NY, USA, 2004. ACM. 138

[71] Wei Liu and Shih-Fu Chang. Robust multi-class transductive learning with graphs. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 381–388, 2009. 44, 99, 108, 129, 131

[72] Zhu Liu, David Gibbon, Eric Zavesky, Behzad Shahraray, and Patrick Haffner. A fast, comprehensive shot boundary determination system. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1487–1490, July 2007. 2, 12

[73] Alexander Loui, Jiebo Luo, Shih-Fu Chang, Dan Ellis, Wei Jiang, Lyndon Kennedy, Keansub Lee, and Akira Yanagawa. Kodak's consumer video benchmark data set: concept definition and annotation. In *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 245–254, New York, NY, USA, 2007. 54

[74] David G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999. 15

[75] Huanbo Luan, Yantao Zheng, Shi-Yong Neo, Yongdong Zhang, Shouxun Lin, and Tat-Seng Chua. Adaptive multiple feedback strategies for interactive video search. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 457–464, New York, NY, USA, 2008. ACM. 40, 93

[76] Michael I. Mandel, Graham E. Poliner, and Daniel P.W. Ellis. Support vector machine active learning for music retrieval. *Multimedia Systems (2006)*, 12(1), August 2006. 4, 14

[77] Paul Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, pages 374–388, 1976. 14

[78] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, 2004. 15

[79] Apostol Natsev, Alexander Haubold, Jelena Tešić, Lexing Xie, and Rong Yan. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 991–1000, New York, NY, USA, 2007. 34, 91

[80] Apostol Natsev, Milind Naphade, and Jelena Tešić. Learning the semantics of multimedia queries and concepts from a small number of examples. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 598–607, New York, NY, USA, 2005. ACM. 31

[81] Shi-Yong Neo, Huanbo Luan, Yantao Zheng, Hai-Kiat Goh, and Tat-Seng Chua. Visiongo: bridging users and multimedia video retrieval. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 559–560, New York, NY, USA, 2008. 39

[82] Giang Phuong Nguyen and Marcel Worring. Optimization of interactive visual-similarity-based search. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(1):1–23, 2008. 36, 93

[83] Nielsen Online. A2/m2 three screen report. http://en-us.nielsen.com/main/news/news_releases/2009/February/tv_internet_and_mobile, February 23, 2009. Accessed: September 15, 2009. 1

[84] Nielsen Online. Nielsen announces september u.s. online video usage data. http://en-us.nielsen.com/main/news/news_releases/2009/october/nielsen_announces, October 12, 2009. Accessed: December 14, 2009. 2

[85] Antti Oulasvirta, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile hci. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 919–928, New York, NY, USA, 2005. 70

[86] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, 20(3):21–36, May 2003. 110, 130

[87] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *CICLing*, pages 241–257, 2003. 55

[88] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 15, 107

[89] Jeremy Pickens, Gene Golovchinsky, Chirag Shah, Pernilla Qvarfordt, and Maribeth Back. Algorithmic mediation for collaborative exploratory search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, New York, NY, USA, 2008. ACM. 3, 37, 138

[90] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980. 14, 55

[91] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989. 20

[92] Ran Raz. On the complexity of matrix product. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 144–151, New York, NY, USA, 2002. ACM. 122

[93] Kerry Rodden, Wojciech Basalaj, David Sinclair, and Kenneth Wood. Does organisation by similarity assist image browsing? In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 190–197, New York, NY, USA, 2001. ACM. 92, 104

[94] Yong Rui, Thomas S. Huang, Sharad Mehrotra, and Michael Ortega. A relevance feedback architecture for content-based multimedia information retrieval systems. In *CAIVL '97: Proceedings of the 1997 Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '97)*, page 82, Washington, DC, USA, 1997. IEEE Computer Society. 38

[95] Toshio Sato, Takeo Kanade, Ellen K. Hughes, Michael A. Smith, and Shin'ichi Satoh. Video ocr: indexing digital new libraries by recognition of superimposed captions. *Multimedia Syst.*, 7(5):385–395, 1999. 14

[96] Behzad Shahraray and David Gibbon. Automated authoring of hypermedia documents of video programs. In *MULTIMEDIA '95: Proceedings of the third ACM international conference on Multimedia*, pages 401–409, New York, NY, USA, 1995. ACM. 14

[97] Ryan Singer. Just one slider, but not so simple - signal vs. noise (by 37signals). http://www.37signals.com/svn/archives2/just_one_slider_but_not_so_simple.php, May 20, 2005. Accessed May 20, 2008. 60

[98] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29, New York, NY, USA, 1996. 21

[99] Alan Smeaton, Colum Foley, Daragh Byrne, and Gareth Jones. Mobile, ubiquitous information seeking, as a group:the ibingo collaborative video retrieval system. In *MobiQuitous 2008 - The 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2008. 37, 138

[100] Alan Smeaton, Paul Over, and Aiden R. Doherty. Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, In Press, Corrected Proof:–, 2009. 12

[101] Alan Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 321–330, New York, NY, USA, 2006. 25, 117

[102] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000. 11

[103] John Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98, New York, NY, USA, 1996. ACM. 34, 104

[104] John Smith, Milind Naphade, and Apostol Natsev. Multimedia semantic indexing using model vectors. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 445–448, Washington, DC, USA, 2003. 45

[105] Cees G. M. Snoek, Marcel Worring, Bouke Huurnink, Jan C. van Gemert, Koen E. A. van de Sande, Dennis C. Koelma, and Ork de Rooij. MediaMill: Video search using a thesaurus of 500 machine learned concepts. In *Proceedings of the 1st International Conference on Semantic and Digital Media Technologies*, Athens, Greece, December 2006. 18, 54

[106] Cees G. M. Snoek, Marcel Worring, Dennis C. Koelma, and Arnold W. M. Smeulders. Learned lexicon-driven interactive video retrieval. In H. Sundaram et al., editors, *Proceedings of the International Conference on Image and Video Retrieval, CIVR 2006, Tempe, Arizona, July 13-15, 2006*, volume 4071 of *LNCS*, pages 11–20, Heidelberg, Germany, July 2006. 45

[107] Daniela Stan and Ishwar K. Sethi. eid: a system for exploration of image databases. *Inf. Process. Manage.*, 39(3):335–361, 2003. 92

[108] Markus Stricker and Markus Orengo. Similarity of color images. *Storage and Retrieval for Image and Video Databases III (SPIE)*, 2420:381–392, 1995. 14

[109] Jaime Teevan, Eytan Adar, Rosie Jones, and Michael Potts. History repeats itself: repeat queries in yahoo's logs. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 703–704, New York, NY, USA, 2006. ACM. 136

[110] Jelena Tešić and John Smith. Semantic labeling of multimedia content clusters. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1493–1496, July 2006. 37, 54

[111] Qi Tian, Baback Moghaddam, and Thomas S. Huang. Visualization, estimation and user-modeling for interactive browsing of image libraries. In *CIVR '02: Proceedings of the International Conference on Image and Video Retrieval*, pages 7–16, London, UK, 2002. Springer-Verlag. 92

[112] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM. 39

[113] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM. 93

[114] Antonio Torralba, Rob Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 100, 104

[115] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, 1991. 104

[116] Mark R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55(2-3):71–82, 1986. 15

[117] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 16

[118] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:511, 2001. 2

[119] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, New York, NY, USA, 2004. ACM. 17

[120] Avery Wang. An industrial-strength audio search algorithm. In *Proceedings of the Fourth International Conference on Music Information Retrieval*, Baltimore, MD, October 2003. 2, 14

[121] Jun Wang, Shih-Fu Chang, Xiaobo Zhou, and S. Wong. Active microscopic cellular image annotation by superposable graph transduction with imbalanced labels. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 44, 100, 104, 108, 129

[122] Jun Wang, Yu-Gang Jiang, and Shih-Fu Chang. Label diagnosis through self tuning for web image search. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, June 2009. 99, 129, 131

[123] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In Todd K. Leen, Thomas G. Dietterich, Volker Tresp, Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *NIPS*, pages 682–688. MIT Press, 2000. 110, 113

[124] Douglas Wolk. Future of open source: Collaborative culture. http://www.wired.com/dualperspectives/article/news/2009/06/dp_opensource_wired0616, June 15, 2009. Accessed: September 5, 2009. 3, 37, 88

[125] Marcel Worring, Cees G. M. Snoek, Ork de Rooij, Giang P. Nguyen, and Arnold W. M. Smeulders. The MediaMill semantic video search engine. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages –, Honolulu, Hawaii, USA, April 2007. 3, 47, 48

[126] Lei Wu, Xian-Sheng Hua, Nenghai Yu, Wei-Ying Ma, and Shipeng Li. Flickr distance. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 31–40, New York, NY, USA, 2008. 33, 90, 94

[127] Songhua Xu, Hao Jiang, and Francis C.M. Lau. Personalized online document, image and video recommendation via commodity eye-tracking. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 83–90, New York, NY, USA, 2008. 41, 69

[128] Rong Yan, Jun Yang, and Alexander Hauptmann. Learning query-class dependent weights in automatic video retrieval. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 548–555, New York, NY, USA, 2004. ACM. 33, 52

[129] Akira Yanagawa, Shih-Fu Chang, Lyndon Kennedy, and Winston Hsu. Columbia university's baseline detectors for 374 lscom semantic visual concepts. Technical Report 222-2006-8, Columbia University, March 2007. 7, 20, 22, 54

[130] Yiming Yang, Jaime G. Carbonell, Ralf D. Brown, and Robert E. Frederking. Translingual information retrieval: learning from bilingual corpora. *Artif. Intell.*, 103(1-2):323–345, 1998. 28

[131] Peter N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA '93: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 311–321, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics. 104

[132] Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Fifteenth ACM International Conference on Information and Knowledge Management (CIKM)*, pages 102–111, 2006. 79

[133] Jie Yu and Qi Tian. Learning image manifolds by semantic subspace projection. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 297–306, New York, NY, USA, 2006. ACM. 93

[134] Eric Zavesky and Shih-Fu Chang. Cuzero: embracing the frontier of interactive visual search for informed users. In *MIR '08: Proceeding of the 1st ACM international conference on Multimedia information retrieval*, pages 237–244, New York, NY, USA, 2008. ACM. 7

[135] Eric Zavesky, Shih-Fu Chang, and Cheng-Chih Yang. Visual islands: intuitive browsing of visual search results. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 617–626, New York, NY, USA, 2008. ACM. 36, 92, 138

[136] Eric Zavesky, Zhu Liu, David Gibbon, and Behzad Shahraray. Searching visual semantic spaces with concept filters. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 329–336, Washington, DC, USA, 2007. IEEE Computer Society. 24, 58

[137] Dong-Qing Zhang and Shih-Fu Chang. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 877–884, New York, NY, USA, 2004. ACM. 3, 104

[138] Kai Zhang, James T. Kwok, and Bahram Parvin. Prototype vector machine for large scale semi-supervised learning. In *ICML '09: Proceedings of the 26th Annual*

International Conference on Machine Learning, pages 1233–1240, New York, NY, USA, 2009. ACM. 104, 112, 125

[139] Dengyong Zhou, Olivier Bousquet, Thomas N. Lal, Jason Weston, Bernhard Schölkopf, and Bernhard S. Olkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, volume 16, pages 321–328, 2003. 44, 99, 100, 101, 103, 104, 108, 134

[140] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003. 44, 99, 100