MULTIVIEW RANK LEARNING FOR MULTIMEDIA KNOWN ITEM SEARCH

by

David L Etter

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Computer Science

Committee:

_____	Dr. Carlotta Domeniconi, Dissertation Director
_____	Dr. Daniel Barbara, Committee Member
_____	Dr. Zoran Duric, Committee Member
_____	Dr. Siddhartha Sikdar, Committee Member
_____	Dr. Sanjeev Setia, Department Chair
_____	Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering
Date: _____	Spring Semester 2015 George Mason University Fairfax, VA

Multiview Rank Learning for Multimedia Known Item Search

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

David L Etter
Master of Science
Hood College, 2001
Bachelor of Arts
Shippensburg University, 1994

Director: Dr. Carlotta Domeniconi, Professor
Department of Computer Science

Spring Semester 2015
George Mason University
Fairfax, VA

Dedication

I dedicate this dissertation to my wife Lori and my children Rachel, Alaina, and Melanie. Thank you for your sacrifice, love, and support on this journey that we have made together.

Acknowledgments

First, I would like to thank my family for all of their support. Thank you to my parents Donald and Linda who raised me with love and are always there when I need them. Thank you to my grandparents Wilber and Mossie who provide the example that we all strive to achieve. Thank you to Jim and Pauline for always putting their family first.

Thank you to my advisor Dr. Carlotta Domeniconi who guided me in my research and was always patient through the ups and downs of this journey. Thank you to my committee members, Dr. Daniel Barbara, Dr. Zoran Duric, and Dr. Siddhartha Sikdar, whose advice helped me to focus my research.

Table of Contents

List of Tables

List of Figures

Abstract

MULTIVIEW RANK LEARNING FOR MULTIMEDIA KNOWN ITEM SEARCH

David L Etter, PhD

George Mason University, 2015

Dissertation Director: Dr. Carlotta Domeniconi

Known Item Search (KIS) is a specialized task of the general multimedia search problem. KIS describes the scenario where a user has seen a video before, must formulate a text description based on what he remembers, and knows that there is only one correct answer. The KIS task takes as input a text-only description and returns the ranked list of videos most likely to match the known item.

A KIS query is a verbose text description which is used to search a video repository consisting of metadata, audio, and visual content. The task presents a challenge in mapping the unique views of the video and query into a common feature space for search and ranking. Additionally, the queries often include key terms or phrases which are mapped into an incorrect multimedia view. The mapping problem causes the result set to drift away from the intended meaning of the original query. Supervised learning approaches to the KIS problem must overcome the imbalance of positive to negative examples that results from having a single known item.

We introduce a multiview rank learning approach to KIS, based on boosted regression trees, which provides a common feature space and overcomes the view ranking challenge. Natural language processing techniques are used to address the view drift problem by extracting key phrases from the original query which align with a specific video view. This

approach allows us to activate only those views of the video which are applicable to the given query. A semi-supervised rank learning approach is used to overcome the class imbalance of having a single known item. Pseudo-positive examples are identified in a similarity graph and a K-Step Markov approach is used to estimate the importance of nodes relative to the truth root node. We evaluate our approach using benchmark datasets from the TRECVid evaluation [1] and a large social media collection.

Chapter 1: Introduction

1.1 Known Item Search

The convenience of smart phones with high quality video capture capability has led to an explosion in the size of personal and internet video collections. Consumers now use their phones to capture and share short clips of personal activities, news events, blogs, and how-to instructions. According to YouTube press [2] over 100 hours of video are uploaded each minute and over 6 billion hours of video are watched each month. As the volume of content in these repositories expands, there is an increased need for effective multimedia search which exploits the multiple modalities of a video.

Known Item Search (KIS) is a specialized task of the general multimedia search problem. KIS describes the scenario where a user has seen a video before, must formulate a text description based on what he remembers, and knows that there is only one correct answer. As an example, consider the TRECVid 2012 [1] KIS topic 1213, "Find the video of a round silver colored weather satellite, men in white hard hats, nose cone placed in rocket, and rocket lifting off". Table 1.1 displays the known item for the satellite topic and provides examples from the visual, audio, and metadata content. The KIS task takes as input a text only description and returns a ranked list of videos, most likely to match the known item.

Figure 1.1 provides an overview of the multimedia search process. The multimedia search process begins with a text query and a collection or repository of multimedia objects. A multimedia object consists of visual, audio, and metadata content. The visual content includes the images, video, and still frames within the multimedia object. Audio content includes the speech and sound data from a video. Metadata includes text which describes the content of the video or image. Throughout this thesis will we often refer to the full multimedia object as a video.

Table 1.1: Known Item from TRECVid 2012

Video Frames	  
File Name	$1959-02-19_Weather_Eye.\_o\_.1959-02-19_Weather_Eye_512kb.mp4$
Optical Character Recognition (OCR)	rwx EYE Vanguard Satellite Suns Sky From Space Hum
Title	Weather Eye Vanguard II Satellite Scans Sky From Space 1959 02 19
Description	Vanguard II satellite placed in nose cone of rocket launched partial newsreel brief silence at start of story
Automated Speech Recognition (ASR)	Satellite electric eyes will scan the Earth s cloud cover broadcasting another reporting of the weather stations...space vehicle one of the most technically sophisticated of the space rockets misfortune six times...

The individual components of the multimedia content are defined as modalities or views. The visual content includes views for motion data, optical character recognition (OCR), and low level features based on color, texture, and shape. The audio content can be used to defines views for sound and automated speech recognition (ASR). Metadata views can include location information, text description, titles, keywords, and comments.

We search the repository using a text only query, or topic, which is often a verbose description that includes phrases identifying one or more views. The query and multimedia views are mapped into a common feature space which provides a low level representation

that can be used by our ranking algorithm. The output of the search is a ranked list of multimedia objects based on their relevance to the given input query.

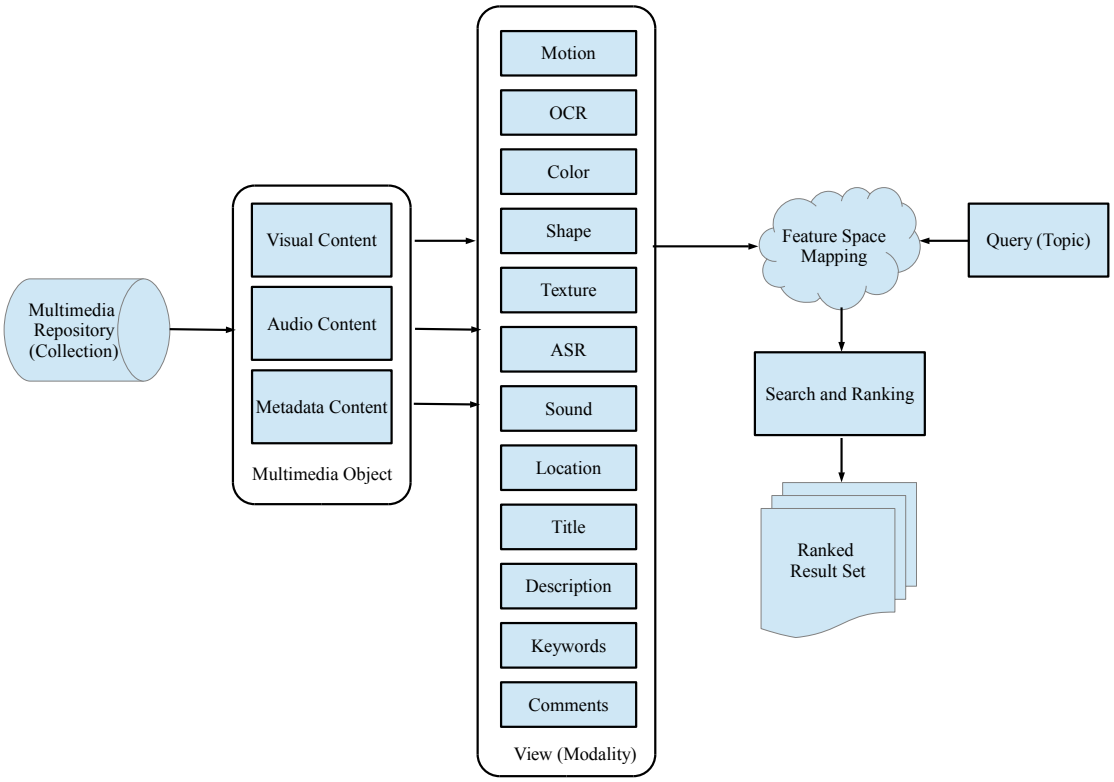


Figure 1.1: Multimedia Search

1.2 Challenges

Video search and ranking approaches can be divide into three categories [3]. The first category is query-by-keyword, where traditional information retrieval models are used to match a text query with the video metadata. This approach generally ignores the image content of the video. The second category is query-by-example, which includes sample images as part of the query. This approach uses both text and visual modalities, but forces a user to provide sample images. The final category is query-by-concept, where the video is tagged with semantic concept labels and the user provides a list of concepts as the query.

Research on the KIS task has generally followed a query-by-keyword approach, since only a text query is provided. State of the art approaches to the KIS task can be categorized as query analysis and result fusion. Query analysis approaches have focused on classifying query keywords based on modality [4] and expanding multimedia content through external knowledge bases [5]. Fusion approaches have been successful by using a late linear fusion model learned over similar query clusters [6].

These approaches to KIS attempt to overcome the primary challenge of the task, which is how to map the unique views of the multimedia object and query into a common feature space for search and ranking. We decompose this problem into the three challenges defined as view ranking, view drift, and class imbalance.

1.2.1 View Ranking

View ranking identifies the challenge of how to effectively model the search and ranking problem over a set of views derived from the metadata, audio, and visual content. Metadata includes views from donor content such as filename, title, description, subject, and keywords. This content is often incomplete or missing, varies in length, and include numerous misspellings. ASR and OCR are audio and visual views that can be mapped into a text feature space, but are often noisy and incomplete. Low level visual features, such as color, texture, and local keypoint can be extracted from the video content, but again it is not clear how to map the text topic request to these feature spaces.

1.2.2 View Drift

View drift defines the problem where key terms or phrases within a query are mapped into the incorrect multimedia view. The mapping problem causes the result set to drift away from the intended meaning of the original query. This is a similar problem to the query-drift [7] that is often found in the information retrieval community. A baseline approach for video retrieval is to submit the full query to an index composed of the text representation of the videos. In this scenario, all of the views such as metadata, visual, automated speech recognition, and optical character recognition, are concatenated into a single view. The problem with this approach is that the influence of phrases which are more selective for a single view are diminished in the concatenated view.

1.2.3 Class Imbalance

Class imbalance is one of the challenges for a supervised learning approach on the KIS problem. The goal of the KIS task is to return a ranked list of the 100 videos most likely to match a given query. The results set for any given query, consists of 99 negative examples and only 1 positive example. This imbalance of positive to negative training examples presents a challenge to any supervised learning algorithm. In addition, while only one known item is correct for a given query, we find that many videos share similar content. These similar videos result in negative training examples that may share a similar feature space to that of the single known item.

1.3 Contributions

Our contributions to the multimedia retrieval community include the following:

1. We introduce a multiview rank learning approach for multimedia KIS to overcome the view ranking challenge. In our model, each of the views derived from the metadata, audio, and visual content are treated as a unique view within the system. We model

the individual feature space for each multimedia view and create a view specific ranking model using gradient boosted regression trees. This algorithm uses a hierarchical model, where the output of view specific models are combined into a final multiview ranking. The hierarchical approach allows the initial view specific models to focus on their own unique feature space before attempting to merge results. Each view includes a unique description of the feature space which may be difficult to capture in a single ranking model.

2. To address the view drift problem we identify and extract key phrases from the original query which align with a specific video view. This approach allows us to identify and activate only those views of the video which are applicable to the given query. Natural language processing techniques, such as named entity extraction and dependency tree parsing, are used to identify the key phrases from the original query. We introduce a supervised rank learning algorithm to construct a model for identifying the correct view mappings. The model output on a set of previously unseen queries allows us to select N phrases from the ranked list and activate only those views relevant to the given phrase.

3. We introduce a semi-supervised rank learning approach to overcome the class imbalance problem found in the KIS task. Our algorithm identifies pseudo positive training examples from each of the multimedia views. This semi-supervised approach uses a ground truth video to identify similar videos in each of the individual modalities. To identify pseudo examples we model the similarities as a graph and use a K-Step Markov approach to estimate the importance of nodes in the graph relative to the truth root node. Each pseudo positive example is then assigned a decreasing graded relevance based on the distance from the truth video. This approach allows us to include visual, audio and metadata views when identifying pseudo positive examples.

4. We present the first evaluation of KIS on a social media collection. Our evaluation set consists of approximately 250,000 multimedia objects collected from the public feed of

a major social media site. We identify the challenges presented by this diverse set of image and video content. These challenges include the use of informal text throughout the social media metadata content. We present an NLP approach to overcome the challenges of informal text and show that our multiview ranking algorithm is effective for this domain.

1.4 Evaluation

We evaluate our approach using two large multimedia data sets. The first, is the benchmark data and truth sets from the TRECVid 2010-2012 [1] Known Item Search task. Experiments are conducted using a set of approximately 25,000 internet videos, licensed through Creative Commons. The videos total approximately 600 hours and range in duration from 10 seconds to 3.5 minutes. The content of the videos cover a wide range of topics including short documentaries, home videos, and commercials. The videos includes metadata, provides by the donor, in the form of filename, title, subject, keywords, and description. The evaluation set includes approximately 1,000 queries derived from a set of videos drawn at random from the internet video archive.

Our second data set consists of approximately 250,000 multimedia objects collected from the public feed of a major social media site. Social media content includes photos and videos from daily life activities, inspirational messages, advertisements, art, leisure, travel, sports, and entertainment. The content also includes metadata such as geographic location information, text captions, hashtags, and comments. The evaluation set includes approximately 300 queries derived from a random selection of the social media collection.

The KIS task goal is for the system to return a ranked list of the top 100 multimedia objects most likely to match the query. Mean inverted rank is used to measure the performance of the known item system.

1.5 Organization

The remainder of this thesis is organized as follows:

Chapter 2 introduces background material and related work. We begin with a discussion of the challenges associated with multimedia retrieval and then discuss the four main components of a retrieval system.

Chapter 3 provides our first look at the multimedia KIS task. The chapter discusses some of the challenges with search and ranking over multimedia modalities such as audio, metadata, and visual content. We introduce a semi-supervised rank learning approach to overcome the KIS modality gap. This semi-supervised approach uses a ground truth video to select similar videos in each of the individual modalities. We then model the similarities as a graph and use a K-Step Markov approach to estimate the importance of nodes in the graph relative to the truth root node.

Chapter 4 introduces *Multi*^{2}*Rank*, our multiview learning algorithm for KIS. The multiview approach defines a unique view and feature space, for each modality of the multimedia object. We uses natural language processing techniques to identify view specific phrases and output a ranked mapping of the phrases into their respective views. Next, we model the individual feature space for each multimedia view and create a view specific model using gradient boosted regression trees. Our approach is evaluated on a benchmark TRECVid dataset and achieves state of art performance for this KIS task.

Chapter 5 presents our query-by-concept approach for multimedia retrieval where both the query and video are mapped into a set of semantic concepts. This chapter focus on the problem of mapping a KIS text query into a set of semantic concepts. We present an approach which minimizes concept drift by first extracting key phrases from the query text using natural language processing. A semantic ranking model is used to identify the candidate set of query-concept pairs. We evaluate our approach using a set of KIS queries and truth concepts from the TRECVid evaluation.

Chapter 6 continues our study of query-by-concept from the perspective of the multimedia repository. We propose a semantic rank learning model, called SemRank, to overcome

the challenges of limited vocabulary size and lack of training data. A semantic fusion model is used to combine the output from many noisy classifiers. The approach is evaluated over a large internet video repository, and the results show that query-by-concept can be an effective model for multimedia KIS.

Chapter 7 expands the domain of the multimedia KIS problem to social media. The chapter examines the challenges of metadata and visual content posed by social media. We model this unique feature space with our multiview ranking and evaluate the approach on a massive data set of images, video, and metadata.

Chapter 8 summarizes this work and highlights our contributions to the multimedia retrieval community. We also explore new research directions and pose a number of questions for future multimedia retrieval research.

Chapter 2: Background and Related Work

The accuracy, reliability, and usability of today's text-html based search engines have revolutionized the way people consume data. Finding the phone number for a pizza shop or the latest news on your favorite football player is available quickly and accurately from any one of dozens of open or commercial web search engines. The same success has not been achieved by current video search engines. The majority of commercial video search engines base their search ranking on text labels provided by human analysis or from the context of text that surrounds the data. The inability to directly search in the feature space of the video data has resulted in both poor precision and recall.

The retrieval of video data is unique in that it is composed of multiple modalities, such as sound, speech, text, vision, and motion data. Recent research in image analysis and automatic speech recognition has lead to significant advances in the ability to model these unique feature spaces. However, there are a number of challenges that impede the progress towards robust video retrieval systems. The question of how to effectively fuse the output from each of the uni-modal feature spaces into a final ranked result remains an open research problem. A second challenge is the ability to bridge the gap between a user information need, in the form of a text query, and the feature space of the video data. This problem is often referred to as the semantic gap. We believe that these two problems are not independent and any solution must consider both query analysis and multi-modal fusion.

2.1 Multimedia Retrieval

A multimedia retrieval system includes four major components: feature extraction, query analysis, search, and result fusion. Feature extraction maps each of the metadata, audio,

and visual content views into a feature space representation. The query analysis component provides a mapping of the text query to a feature space. The search and ranking algorithm returns a ranked result list, for each view, of the videos most likely to match the given query. Finally, result fusion merges the results from the different view rankings.

2.1.1 Feature extraction

Multimedia data is unique in that it is composed of multiple modalities, such as sound, speech, text, vision, and motion data. Current approaches to the feature extraction component borrow from the fields of vision and speech recognition. Feature extraction in video processing describes the process of extracting structural information from the video data and providing a compact feature vector which uniquely describes that information. This representation makes up the feature space and provides the basis for search and classification of the video stream.

Low-level visual features are structures which help describe the color, texture, and shape of an image. These features are extracted from the visual content at the pixel level and can be used to index, browse, and search image and video data. Features can be extracted and described at global, region, local, or grid levels within the visual data.

The MPEG-7 Visual standard [8] defines a set of low-level visual descriptors which can be used to describe and measure similarity in image and video data. The standard includes a set of general visual descriptors for color, texture, shape, and motion features. The color features describe the color distributions within an image and include descriptors for spatial color, dominant color, and color layout. Texture features describe the visual patterns and edges that are found in an image and are defined in the standard using homogeneous texture and edge distribution descriptors. The shape features describe objects in an image through its contours or spatial distributions and the motion descriptors define object and camera motion vectors.

The edge direction histogram [9] which is part of the MPEG-7 standard has become

one of the dominant low-level features used in video retrieval because of its compact representation, computational efficiency, and its invariance to image translation and rotation. The descriptor divides the image into 16 sub-images, using a (4*4) grid, to allow for the calculation of localized edge distributions. The edge histogram consists of four directional edges and one non-direction edge for each of the sub-images. The complete image is represented by an 80-dimensional feature vector, where each dimension contains the normalized bin count for one of the five edge types in each of the 16 sub-images. Image search and matching can be performed using the local edge features or by combining different local segments to produce a global or semi-global feature space.

Local key point features have recently attracted significant interest in the video retrieval community [10] [11]. Keypoint detectors attempt to detect a small set of locally stable points and their surrounding region. These keypoints are then clustered into a set of visual words to form the visual vocabulary for a keyframe and form the basis of the bag-of-features representation. The size of the visual word vocabulary and the weight of each term is an important parameter in the bag-of-features representation. One of the problems with this model is that two keypoints assigned to the same cluster may differ in their distance to the cluster center. A soft-weighting scheme was proposed in [10] to overcome this problem by considering the visual words which are the top-N nearest neighbors of a keypoint.

The use of low-level visual feature extraction in video retrieval raises several issues related to the amount of images which need to be processed and the acceptable level to which those features describe the data. In the case of the TRECVID evaluation, systems must be able to extract features from over 200 hours of video data. The ability to process and search large volumes of video data dictates the use of features which can be extracted quickly and represented in low dimensional vectors. The use of faster and lower dimensional feature spaces often means making a tradeoff for speed over accuracy.

High-level features are the semantic concepts that we use to describe objects, events, and activities. An example of a high level concept would be a beach scene. This concept could be further broken down into sub-concepts such as sand, water, people, rocks, sun, and

sky. These high-level concepts are a closer reflection of how humans express an information need and provide an option to help bridge the semantic gap. A common approach in video retrieval is to train a classification model to recognize a high-level concept using low-level visual features [12]. The model can then be used to assign the probability that a high-level concept exists in a previously unseen video shot.

Table 2.1: LSCOM Concepts

ID	Name	Definition
000	Parade	Multiple units of marchers, devices, bands, banners or Music
001	Exiting Car	A car exiting from somewhere, such as a highway, building, or parking lot
002	Handshaking	Two people shaking hands. Does not include hugging or holding hands
003	Running	One or more people running
004	Airplane Crash	Airplane crash site
005	Earthquake	Wreckage from an Earthquake
006	Demonstration Or Protest	One or more people protesting. May or may not have banners or signs
007	People Crying	One or more people with visible tears
008	Airplane Takeoff	Airplane heading down the runway for take off (may have already left runway and be ascending)
009	Airplane Landing	Airplane descending or decelerating after making contact with runway

The Large Scale Concept Ontology for Multimedia (LSCOM) [13] is a project which brought together research in the areas of retrieval, machine learning, and knowledge representation, in order to define a set of general high-level concepts. The project goal was to

define a standard set of concepts which could be used for machine tagging of video data. The target domain for the project was based on broadcast news video in cooperation with the TRECVID Evaluation [1]. This work produced an ontology of 1000 concepts and a manual annotation of 450 of these concepts in 80 hours of broadcast news video. A sample of the LSCOM concepts and their definitions are found in Table 2.1.1.

Speech recognition text [14] has played a significant role in the quality of video search engines. The use of speech output allows video search researches to reuse proven text indexing and matching algorithms from the field of information retrieval. Current research includes the use of a vector space model [12] [11] [15] for speech text and the use of inverted indexes. The major drawbacks of this feature are that the quality of the output from speech recognition software is not consistent and that the speech in a video does not necessarily coincide with this visual content. An example of this would be in news broadcast video where an anchor sitting at a news desk describes a story without showing related images or captions.

2.1.2 Query analysis

Query analysis describes the transformation of a query into a feature space which is compatible with that of the video data. A query can be defined as the expression of a user information need and can be posed to a retrieval system in the form of a text description or an example image. Queries posed in the form of example images can be mapped into low-level image features and directly queried against the feature space of the video vision data. Text queries can be matched to speech transcripts using a traditional vector space model or classified into semantic concepts and matched to high-level vision concept features. Extensions to this model include the use of query stop word elimination and query expansion through a dictionary or thesaurus.

Research in query analysis has generally followed a model of expansion or reduction. Query expansion attempts to increase the generality of the text topic by expanding the query using N-grams, ontology's, or pseudo-relevance feedback. Query reduction models

map the query into a reduced feature space in order to discover topics or identify relevant semantic concepts.

Knowledge bases such as WordNet [16], Wikipedia [17], and the World Wide Web provide a rich feature set of semantic information which can be used during query analysis. Automatic term categorization is studied in [18] using the web as a knowledge base to model the context of terms belonging to a given class. The context is obtained using document excerpts, known as a snippets, returned by a web search engine. Snippets provide the context for a query class and are used to train a classifier to predict the label of an unseen query. Terms from the snippets form the basis of an Entity Context Lexicon (ECL) which captures term and snippet frequency for a given entity. Labeled ECL examples are combined to build a Class Context Lexicon (CCL) which can be used to predict the class of an unknown query entity. A semantic kernel is introduced in [19] for text classification which enhances a document representation using semantic knowledge derived from Wikipedia. The semantic kernel utilizes the concepts, categories, and associative relationships defined within the encyclopedia. Each Wikipedia title identifies a concept and is associated with one or more categories. Title redirects define a equivalency relationship between concepts and categories define a hierarchical relationship between concepts. Associative relationships are mined from the article body using the embedded input and output links to related concepts. Given a document, a candidate set of concepts is selected based on term vector similarity to the set of Wikipedia articles. The original document is expanded to include the candidate set concepts, synonyms, hyponyms, and associations. A proximity matrix is constructed which captures the candidate concepts and their relationships and is used to derive the enriched representation of a document.

The work in [20] performs an object-sensitive query analysis in which they try to identify and emphasize targeted objects in a query. A target object is identified in a query by first tagging the nouns and then mapping them to a semantic concept ontology. The thought is that a visual query focuses on one or more specific objects and that these terms should be emphasized. A probabilistic weighting scheme, based on the Best Match (BM25) text

weight algorithm [21], is presented which emphasizes the visual importance of a query term using the target objects.

The automated search task of the TRECVID evaluation provides query topics in the form of a text description and a small set of example images. The sample images are generally treated in a query-by-example model where low-level visual features are extracted and matched with the low-level features of the video repository. The ranked results from the example image search are then fused with the other search modalities. An alternative approach, which follows the concept-based model, is to build a semantic concept feature vector for each example image using a set of high-level concept detectors. This feature vector includes a dimension for each concept which contains the probability that the shot includes that concept. An extension to this approach is presented in [22] where semantic concept feature vectors from sample images are used to train multiple SVM models. Since the images contain only positive examples, pseudo-negative examples are generated by random sampling of the data. Multiple models are created by reusing the positive examples along with different sets of negative random samples.

2.1.3 Search

Multimedia retrieval can be divided into three categories [80] based on the query approach. Query-by-keyword is the traditional information retrieval approach where the text query is compared to the metadata of a repository. Query-by-example uses example images to match against the visual content of a repository. Query-by-concept is the approach where both query and content are mapped into a semantic concept space before comparison.

Query-by-keyword defines the retrieval approach where a text only query is compared to the metadata or speech to text of a multimedia repository. Text search in the form of metadata and speech transcripts or Automated Speech Recognition (ASR) [14] forms one of the base modalities for video retrieval system. The metadata can include multiple modalities such as speech to text, optical character recognition, image features, and descriptive text provided by the video author. Transcripts or ASR can be aligned to shots or

stories within a video and searched using proven techniques borrowed from the information retrieval community. In its simplest form, text within the time bounds of a specific shot is segmented into a visual document and indexed using an inverted index.

Query-by-example describes a search paradigm in which one or more sample images are used as a search query. Color, texture, and shape features are extracted from the example images and compared with the video repository to create a ranked result list. This model has the advantage that its query is expressed in the same rich visual content as the visual data of the video repository. Low-level features such as edge and color are directly matched in their original features spaces. This model produces good results if the images in the database are exact or near matches, however a few specific query images will not scale to a generic concept. To overcome this limitation a topic modeling approach is applied in [23]. They attempt to capture the common low-level features of the example images using a Gaussian mixture model.

Concept based video search has recently emerged as an efficient and viable solution to bridge the semantic gap. This approach attempts to merge the results of a number of high-level concept detectors in order to answer an information need. These concept detectors are generally classifiers, borrowed from the machine learning community, and trained using low-level vision features of positive and negative example images. A unique detector is trained for each high-level concept and their combined results can be pooled to create a semantic description of a video shot. A user query can then be transformed into the semantic space of the concept detectors and a ranked result returned based on the test data semantic descriptions.

The first framework for concept-based search was presented in [24] using a lexicon of 46 concepts with positive training examples manually annotated in the TRECVID 2003 development data set. A Support Vector Machine (SVM) [25] was trained for each of the 46 concepts using positive and negative examples from the development set. The input features consisted of a concatenated vector of low-level images features from color correlogram, co-occurrence texture, edge histogram, and moment invariants. The concept models were then

applied to shots in the test data and a 46 dimensional model vector is created for each shot. Each dimensional value contains the confidence of that concept occurring in the video shot.

The concept based solution raises two important questions; which high-level concepts should be modeled and how can a query be transformed to the semantic concept space. Training a classifier to recognize a high-level concept is not a trivial task. The classifier must have the ability to generalize to different viewpoints, backgrounds, and sizes for the selected concept. This requires numerous positive training examples which are not readily available. The concept based solution must also consider the number of concept detectors that can realistically be evaluated over video repositories which contain hundreds or thousands of hours of video. The second question is how to map query terms which do not have a corresponding detector. In this case the query needs to be mapped to the best set of available detectors whose result sets can then be fused using the weights to each detector in order to estimate the previously unseen concept.

As noted earlier, LSCOM was one of the first attempts by the research community to answer the question of which concepts to model. A subset of these concepts, known as LSCOM-lite [26], was annotated on the 2005 TRECVID evaluation. This subset includes 39 semantic concepts Table 2.1.4 observed in broadcast news video and is intended to cover a diverse semantic space.

2.1.4 Fusion

A semantic concept detector or a Multi-Query-By-Example (MQBE) search on a low-level visual feature can be considered an expert since it takes a query and produces a ranked output. Combining the results of these experts has been shown to boost overall retrieval performance and is an active area of research. The performance gain from merging the different ranked lists occurs because each expert performs better on certain tasks or concepts. The process of merging these different modalities and detector outputs is known as fusion in the video retrieval research community. Fusion includes strategies for combining, weighting, and re-ranking expert results.

Table 2.2: LSCOM-lite TRECVid 2005 Semantic Concepts

Category	Semantic Concept
Location	Office, Court, Meeting, Studio, Outdoor, Road, Sky, Snow, Urban, Waterscape, Mountain, Desert, Building, Vegetation
People	Crowd, Face, Person, Roles, Govt Leader, Corp Leader, Police, Military, Prisoner
Objects	Flag US, Animal, Computer, Vehicle, Airplane, Car, Boat/Ship, Bus, Truck
Activities & Events	People, Walk/Run, March, Explosion Fire, Disaster
Program Category	Weather, Entertainment, Sports
Graphics	Maps, Charts

Fusion strategies are generally classified as either early or late. An early fusion strategy attempts to combine the features vectors from multiple modalities before using a learning algorithm. This strategy has the advantage that the learning algorithm is applied to a single feature vector which represents a combined feature space for the video data. A late fusion strategy applies a separate learning algorithm for each modality and then merges their final outputs. The advantage of this approach is that the learning algorithms are applied over a consistent feature space. Early versus late fusion was studied in [15] using the TRECVID 2004 evaluation data sets. Average precision was measured for 20 semantic concepts using visual and text features. Their results indicated that a late fusion approach generally performed better, but for some concepts the early fusion was significantly better.

A multi-modal fusion model is presented in [27] where a dynamic class is generated for a test query based on a semantic analysis of the query text and co-occurring query features. The retrieval system incorporates text, visual, and model based experts for the creation of multiple unique ranked lists. A final ranked list is generated by a linear combination of results from each expert using a weight vector. The dynamically generated weights are

learned through a query analysis which tags the query with semantic concepts from a large ontology. These semantic tags are then mapped to seven visual categories: Sports, Named-Person, Unnamed-Person, Vehicle, Event, Scene, Others. Each category represents a query dependent class whose weights have been optimized using training examples. Unseen queries are then mapped to the k nearest neighbors of the training queries and weights are optimized using their average performance.

A neighborhood fusion method is considered in [28] to merge results from multiple concept detectors. This method considers both the detectors performance on neighboring shots and the detectors overall average precision when generating the final ranked result. The premise being that if a concept is detected in a shot there is a high probability it should be detected in neighboring shots. A spectral clustering approach is presented in [29] for multiview re-ranking. The initial ranked results from each feature are divided into k distinct clusters using a Normalized Cuts clustering algorithm. The clusters are then ranked using a Hausdorff distance measure and merged with a Cross-Reference (CR) fusion strategy which weights shots based on their combined cluster ranking across features.

A query time coefficient generation is described in [30] where they perform a real-time evaluation of the contribution for different low-level features. The algorithm utilizes the distribution of the shot scores on a Query-By-Example (QBE) to each low-level feature. This approach is based on the theory that a feature which under goes a rapid change in its normalized score provides a better separation for the given query example.

A graph based re-ranking model is proposed in [20] to exploit the semantic concept and low-level feature relationship between shots. The ranking algorithm is based on the intelligent surfer PageRank [31] and defines video shots as vertexes and their relationships as hyperlinks. The relationships are modeled using text search as a baseline with an expansion to semantic concepts and low-level visual features. Edge direction in the graph is determined using the confidence score from the semantic concept detectors.

2.1.5 Evaluation

Precision and recall are the two metrics most commonly used for measuring performance in information retrieval systems [32]. Precision is defined as the number of relevant shots retrieved divided by the total number of shots retrieved. Precision provides a good measure for accessing the relevance of the shots returned by a video retrieval system, but does not provide any information on how many relevant images were not returned by the system. Recall is defined as the number of relevant shots retrieved divided by the total number of relevant shots. Recall provides a good measure of what percentage of relevant shots were retrieved but does not provide information on how many non-relevant shots were retrieved. The complementary nature of these two measures has lead to a number of hybrid metrics which combine the two.

One such measure is Precision at Fixed Recall Levels. In this evaluation measure a specific number of recall points are chosen and the precision is measured at each of these recall points. An example would be the selection of ten recall points corresponding to 10% through 100% of relevant documents retrieved. This measure is best known for its plot on a precision-recall curve. A similar measure is the Precision at Fixed Points, in which the precision is calculated after a fixed number of shots have been retrieved. The fixed point measure is often seen as providing a more meaningful combination of precision recall to an end user than the fixed recall level because its precision points are fixed. R-precision is a variation of the fixed point measure, in which R corresponds to the number of relevant documents and the precision is calculated after R documents are retrieved.

Mean average precision (MAP) provides a single evaluation measure which combines precision and recall. In this measure, precision is calculated at the result rank of every relevant document and then averaged for all relevant documents. Relevant documents that are not retrieved by the system are assigned a precision value of 0. The ability to combine precision and recall into single measure has made average precision the most commonly used metric for information retrieval systems. However, this measure presents a challenge for large benchmark data such as that of the TRECVID collection. The size of this and

other modern day test collections make the ability to identify all relevant shots, for all query topics, a difficult and impractical task. This limitation has lead to the adoption by the video retrieval community of a new evaluation measure, Inferred Average Precision (IAP) [33]. The inferred average precision evaluation measure recognizes average precision as the expected value of a random experiment and attempts to infer average precision of the overall collection from a random sample of shot judgments.

2.2 Related Work

Our work in KIS is influenced by related work in a number of research areas including known item search [1], semantic indexing [34], rank learning [35] [36], multiview learning [37], and information retrieval [38].

2.2.1 Known Item Search

KIS describes the task, where a user has previously seen a video and wants to find it again using a text only description. An example scenario would be that we want to show a friend a video clip about a new game that we recently watched in our favorite online repository. In order to find the video again, we query the repository using a text description of what we remember from the video. This is an example from the TRECVid evaluation [1], "Find the video of an Sega video game advertisement that shows tanks and futuristic walking weapons called Hounds".

Research related to the KIS problem has occurred in multiple text based domains such as person document [39], web [40], email [41], Twitter, and Facebook [39]. Personal document search is studied in [40] over email, presentations, web pages, and pdfs. They investigated techniques for improving document type prediction in personal desktop search. Their model uses type specific meta-data to generate a field-based collection query likelihood. Type specific results are then merged into a final ranked list which improves overall retrieval performance. The results in [39] show that a mixture of language models which combine

evidence from different representations is an effective approach for this type of document retrieval. A KIS task using email data was studied during the TREC 2005 [41] evaluation. The evaluation was conducted over a set of emails taken from the World Wide Web Consortium. Most participants of the evaluation considered both email text and metadata such as anchor text, threads, titles, and dates.

KIS in the context of multimedia search has been studied as part of a TRECVid task [1] in 2010, 2011, and 2012. The video collection for the TRECVid task consists of approximately 8,000 Internet Archive Videos and 300 topics and judgments for each task year. Participants were provided with a text only description of the topics along with the test video collection and metadata. The goal of the evaluation is to return a ranked list of the top 100 videos most likely to contain the topic. A system is scored based on the mean inverted rank (MIR) of where the ground truth video is found.

An overview of the various KIS systems used during the 2011 TRECVid evaluation can be found in [42] and [43]. Task participants attempted to bridge the understanding gap between a topic text and the video collection. Text based approaches included enriching topics and meta-data using external knowledge such as Wikipedia, ontologies, or translations. An approach to bridging the visual modality gap was to identify examples images from a web image search engine. Most of the task participants concluded that the visual modalities provided little benefit to the final rankings. The top scoring team in the task created a classifier that transformed the original text topic into a set of shorter modality specific queries. All of the participating systems attempted to fuse multiple modality results. A query to modality mapping approach for multimedia KIS is studied in [4]. Their approach identifies key phrases for different modalities such as visual, text, and high level features using regular expression, dependency parse, and supervised classification.

2.2.2 Semantic Indexing

Semantic concept indexing is a query-by-concept approach to bridge the semantic gap for multimedia retrieval. In this approach, a set of concepts are used to provide a high level

feature representation for describing objects, events, and activities. Multimedia concept classification systems [34] automatically label a video with a set of high level semantic concepts.

The Large Scale Concept Ontology for Multimedia (LSCOM) [13] was a collaborative effort among researchers to develop a standard set of semantic tags. The work produced a set of 1000 semantic concepts that were used to describe a large collection of broadcast news video. A collaborative annotation on the TRECVid [1] collection is described in [44] [45]. The work of [44] used an active learning approach to filter the candidate set of video frames for annotation. A semi-automatic annotation approach is used in [45], where the system suggests tags for a video frame based on concept dependencies. Both [46] and [10] study the type of visual features and learning algorithms that optimize a semantic classifier. The work of [46] studies the size of the visual vocabulary [47] feature size required for a semantic classifier. Using a supervised learning approach, they found that a vocabulary size of 1024 to 4096 performed the best on a large video repository.

Semantic concept query expansion is studied in [48]. They propose both a rule-based and statistical query expansion approach to identify concepts in a text query. The identified concepts are used to rerank the initial result set. Semantic search is used in the context of event detection in the work of [49]. Their approach maps text queries to concepts using a text language model constructed for each concept. The language model uses a set of documents, retrieved from a web search, to identify words related to the semantic concept.

2.2.3 Multiview Learning

Multiview learning considers the problem of diverse datasets, such as multimedia, which contain unique views within their content. The multiview problem presents a challenge in constructing a model which represents each of the unique feature spaces.

The multiview learning problem has been studied in a number of domains including web classification [50], image classification [51], video search [37], and video semantic concept extraction [52]. A Co-training approach was introduced in [50] for the problem of web page

classification, where the authors consider separate views for the words on a page and the words in hyperlink. A cross feature learning approach is used in [52] for multiview learning in the semantic concept extraction task of TRECVid. The work of [37] attempts to learn a predictive subspace representation for multiple views in a video search/classification task. They consider a text and visual view consisting of a 1894 dimensional text vector and a 165 dimensional color histogram, extracted from a video keyframe.

Multiview image reranking is studied in [53] using a hypergraph-based learning approach. Multiple manifolds are constructed to represent the different visual feature views and then used to learn the different modality weights and final ranking scores. A multiview and multilabel problem is studied in [51] in the context of image classification. Each image can be labeled with multiple concepts such as tree or bicycle and includes multiple image views such as color and texture. Their approach is able to combine multiple image views with a vector-valued multilabel classification.

2.2.4 Rank Learning

Learning to Rank describes a supervised learning approach for constructing ranking models over a set of training data [54] [35]. The training data consists of a query document pair and a relevance judgment. The pair includes a set of partial ordering from traditional information retrieval models such as term frequency inverse document frequency (TFIDF), probabilistic best match (BM25) [38] [21], and language models (LM) [55]. The rank learning model attempts to learn a final ranking using the relevance judgments and partial ordering. Rank learning models based on regression trees have been successfully used by the information retrieval community [56] [57] and provided the base approach for all of the winning teams at the Yahoo! Learning to Rank Challenge [58] [36].

Approaches to learning to rank are typically categorized into [35] pointwise, pairwise, and listwise. The pointwise approach trains a model to predict the exact relevance score for a given document [35] [59], while the pairwise ranking models consider the relative order between pairs of documents [60] [61] [62]. The listwise approach attempts to learn a model

over the entire document list and optimizes a particular information retrieval evaluation metric [54] [57] [36].

Features used in a rank learning model can include query, query-dependent, and query-independent features [63] [64] [58]. Query-dependent features are those that refer to the interaction of a query and document such as a score from one of the traditional information retrieval models. Query-independent refers to a document only feature such as its length or topics, while a query feature is specific to the query itself. Query features are studied in [63] to evaluate their effectiveness on learning to rank models. These features include values such as the number of unique tokens, the number of named entities, and the number of retrieved document categories for a unique query. The LETOR [64] project provides a set of benchmark datasets and evaluation tools for learning to rank research. The datasets allow researchers to compare their ranking algorithms on a benchmark dataset that includes feature vectors, queries, and relevance judgments.

Reranking has been studied in the context of both image search [65], video search, and semantic indexing. The work in [66] improved MAP on a semantic indexing and retrieval task by reranking an initial video shot score, using a model that considers the homogeneity of the video to which it belongs. Automatic video search reranking is studied by [67], where they identified an initial result set using text search, concept detection, and image query-by-example. The top and bottom ranked items were then used as pseudo positive and negative examples to train a test time model to discover co-occurrence patterns. Their use of pseudo examples differ from our approach, which is applied at training time and identifies similar videos using a graded relevance to a ground truth video. A graph reranking approach was used by [20] to improve the initial text search results for a video search task. In contrast, our graph approach is not used to rerank an initial result set, but instead to identify additional pseudo positive examples for training.

2.2.5 Information Retrieval

Information retrieval models provide an approach to ranking a collection of documents given a query. Our work uses a number of traditional information retrieval models including, the vector space model [38], and the probabilistic best match (BM25) [21].

The vector space model [38] represents a query and the document collection in a high dimensional vector or feature space. The document is represented as follows:

$$d = (d_1, d_2, \ldots, d_m)$$

where d_k $(1 \leq k \leq m)$ is the weight of a term in the document. A query to the document collection is then represented as follows:

$$q = (q_1, q_2, \ldots, q_m)$$

The TFIDF score for a query document pair is represented as follows:

$$TFIDF(d, q) = \frac{\sum_{k=1}^{m} d_k \cdot q_k}{\sqrt{\sum_{k=1}^{m} (d_k)^2} \cdot \sqrt{\sum_{k=1}^{m} (q_k)^2}} \tag{2.1}$$

with term weights

$$d_k = q_k = tf \cdot log\frac{N}{df} \tag{2.2}$$

where tf is the Term Frequency (TF) in the document, N is the number of documents in the collection, and df is the number of documents where the term occurs.

The probabilistic best match (BM25) model [21] provides a ranked list of documents according to their relevance to the given query. The BM25 score for a query document pair is represented as follows:

$$BM25(d, q) = \sum_{k=1}^{m} IDF(q_k) \cdot \frac{f(q_k, d) \cdot (z + 1)}{f(q_k, d) + z \cdot (1 - b + b \cdot \frac{d_{len}}{d_{avg}})} \tag{2.3}$$

where $f(q_k, d)$ is the frequency of q_k in d, z and b are free parameters, d_{len} is the length of d, and d_{avg} is the average length of a document. The inverse document frequency (IDF) is represented as follows:

$$IDF(q_k) = log \frac{N - n(q_k) + .5}{n(q_k) + .5} \tag{2.4}$$

where $n(q_i)$ is the number of documents containing q_i.

2.2.6 Gradient Boosted Regression Trees

Our work uses Gradient Boosted Regression Trees [68] as a base rank learning algorithm. Gradient boosting uses a stage-wise approach to generate an ensemble of weak regression tree models that when combined produce a strong classifier. An overview of the approach is described in Algorithm 1.

Algorithm 1: Gradient Boosted Regression Trees

Input: $(x_i, y_i)_{i=1}^n$ training data, M iterations, $L(y, F(x))$ loss function
Output: $F(x)$ model
$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma)$;
for $m = 1$ ***to*** M **do**
  for $i = 1$ ***to*** N **do**
    $r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)}$
  end;
  $h_m(x) =$ RegTree ($\{(x_i, r_{im})\}_{i=1}^n$);
  $\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right)$;
  $F_m(x) = F_{m-1}(x) + \gamma h_m(x)$;
end;
return $F(x)$;

Consider a set of N training examples, $\{(x_i, y_i)\}_{i=1}^{n}$, where x is the input variable and y is the response. Gradient boosting attempts to approximate the function $F^*(x)$, which maps x to y, by minimizing the expected value of the loss function $L(y, F(x))$:

$$F^* = \arg\min_{F} E_{y,x} L(y, F(x)) \tag{2.5}$$

The approach iteratively constructs an approximation of $F(x)$ using:

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x) \tag{2.6}$$

where $h_m(x)$ is the regression tree model generated at iteration m, γ_m is the learned weight associated with that model, and M is the number of iterations. Frequently used loss functions include mean squared error, log-likelihood, and cross entropy loss.

The approach uses numerical optimization in function space and begins with the initialization step:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma) \tag{2.7}$$

At each iteration m, a set of pseudo-residuals are calculated for $i = 1 \ldots n$,

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{2.8}$$

The pseudo-residuals, $\{(r_{im}, x_i)\}_{i=1}^{n}$, are used to train the regression tree model $h_m(x)$.

The regression tree generates a set of partitions $\{R_k\}_1^K$ of the parameter space, where K is number of leaves. At iteration m, the regression tree model is:

$$h_m(x) = \sum_{k=1}^{K} b_{km} I(x \in R_{km}) \tag{2.9}$$

where b_{jm} is the predicted value in the partition R_{jm}.

The learned weight γ_m is calculated:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right) \tag{2.10}$$

The model is updated for iteration m:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{2.11}$$

2.2.7 K-Step Markov

Relationships between entities are often represented as graphs. A common task on a graph, is to estimate the importance of a set of nodes in the graph, relative to a root node. In this work we construct a graph between similar KIS videos and estimate importance using a K-Step Markov approach [69].

A directed graph $G = (V, E)$, is constructed from a set of nodes V and a set of edges E. The ordered pair (u, v), defines the directed edge connecting node u to node v. We assume that the graph has no parallel edges or loops. $P(u, v)$ defines the set of paths between nodes u and v.

The K-Step Markov [69] approach views the graph as representing a first-order Markov chain. The approach generates random walks of fixed length K, beginning at a root node r, and estimates the probability of spending time at any particular node. We define the probability of transitioning from u to v as $p(v|u) = \frac{1}{d_{out}(u)}$

The importance of node t to the root r is defined as:

$$I(t|r) = [Ap_r + A^2 p_r \dots A^k p_r]_t \tag{2.12}$$

where k is a fixed number of steps, A is the Markov transition probability matrix of size

$n \times n$, and p_r is an $n \times 1$ vector of initial probabilities for the root.

Chapter 3: Semi-Supervised Rank Learning

In this chapter we begin to study the problem of how to map the unique views of the multimedia object and query into a common feature space for search and ranking. Metadata can include author content fields such as filename, title, description, subject, and keywords. This content is often incomplete or missing, varies in length, and include numerous misspellings. ASR and OCR are sound and vision features that can be mapped into a text feature space, but are often noisy and incomplete. Visual features, such as color, texture, and local keypoint can be extracted from the video content, but it is not clear how to map the text query to into a common feature space.

We propose a rank learning approach to the KIS view mapping problem. The goal is to learn a ranking function for each query-video pair, which is represented by a feature space derived from queries, videos, and query-video dependent results. Our ranking algorithm is based on gradient boosted regression trees [36] and is trained using a set of query-video pairs with known relevance labels. The output from the model is a ranked list of videos for the given query.

The KIS task also presents a class imbalance problem of positive to negative training examples for a supervised learning algorithm. As an example, for any given training query and its top 100 ranked results, we assign a relevance of one to the known item and zero to the remaining 99 videos. Further analysis of the negative labeled videos shows, while only one known item is correct for a given query, we find that many videos share similar content. To overcome this class imbalance problem, we identify a set of pseudo positive training examples from each of the multimedia modalities. This semi-supervised approach uses a ground truth video to identify similar videos in each of the individual modalities. To identify pseudo examples we model the similarities as a graph and use a K-Step Markov approach [69] to estimate the importance of nodes in the graph relative to the root node.

Each pseudo positive example is then assigned a decreasing graded relevance based on the distance from the truth video. This approach allows us to include both text and visual modalities when identifying pseudo positive examples.

Our contributions to the multimedia retrieval community include the following:

1. We construct a feature space consisting of query specific, query-video dependent, and video specific features, calculated from the metadata, speech, and visual modalities of our text queries and video repository.

2. We introduce the concept of pseudo-positive KIS examples to offset the class imbalance of having a single known item. Pseudo-positive examples are identified in a similarity graph, using a K-Step Markov to estimate the importance of nodes relative to the truth root node.

3. We study a pairwise rank learning approach to the multimedia KIS problem using gradient boosted regression trees.

3.1 Our Approach

We propose a semi-supervised rank learning approach to the multimedia KIS problem. First we define a feature space derived from queries, videos, and query-video dependent results. Next we identify a set of pseudo positive training examples using a similarity graph constructed from the ground truth videos. The pseudo positive examples are used to assign a graded relevance to our query-video training pairs. Finally, a gradient boosted regression tree algorithm [36] is used to learn a ranking model over the training set.

Given a set of known item queries $Q = (q_1, \ldots, q_n)$ where n is the number of queries and the video collection $V = (v_1, \ldots, v_m)$ of size m, We define the KIS task as a mapping:

$$F(q_i, V) = (s^i_{v_1}, \ldots, s^i_{v_m}) \tag{3.1}$$

where q_i is a known item query, V is the video repository, and $s^i_{v_j}$ is the calculated score

for video v_j with respect to query q_i.

Algorithm 2: KIS Learn to Rank

Input: Q KIS query set, V video collection, Q' truth set
Output: $f(x)$ gradient boosted regression tree model
for *each* q *in* Q **do**
    x_q = build-FeatureVector(q,v);
    G_q = build-SimGraph(Q'_q,v);
    I_q = run-KStepMarkov(G_q, Q'_q);
    x'_q = assign-Relevance(x_q, I_q);
return $f(x)$ = train-KISRankLearn(x');

A high level review of our approach is described in Algorithm 2. Given a set of KIS training queries Q, with relevance labels, and a video collection V, we construct a feature vector x_q consisting of the query features, video features, and query-video dependent features for each query-video pair (q_i, v_i). Next, a video similarity graph is constructed, where a truth video Q'_q is the root node of the graph. An importance score is given to each node of the graph using a K-Step Markov approach. The graph node score is used to determine the relevance weight assigned to each query-video pair. The KIS rank learning model is generated using gradient boosted regression trees trained over the relevance weighted feature vector. Given a previously unseen query, we construct a feature vector for each new query-video pair and use the KIS rank learning model to output a final ranking score.

3.1.1 Feature space

To model the KIS feature space, we identify three categories of features which are defined in Table 3.1. Query features are derived from the text of the known item query and include

Table 3.1: Query, Query-Video Dependent, and Video Features

Type	**Feature**	**Description**
Query	Term	Count of terms
	Unique Term	Unique terms
	Person	Count of Person Named Entity
	Location	Count of Location Named Entity
	Organization	Count of Organization Named Entity
Query-Video Dependent	TFIDF	TFIDF Weight Model
	BM25	BM25 Probabilistic Model
	LMIR	Language Model
	Percent Term	Percentage of Term Match
	IDF	Inverse Document Freq of Match
	TF	Term Freq of Match
Video	Term	Count of terms
	Unique Term	Unique terms
	Person	Count of Person Named Entity
	Location	Count of Location Named Entity
	Organization	Count of Organization Named Entity
	Edge	Edge direction histogram
	Color	Color histogram
	Keypoint	Visual Bag-of-words
	Concepts	Semantic concepts

term count, unique term count, and named entity counts [70]. The person, location, and organization named entities are identified using a sequence tagger [71].

Query-Video Features are derived from the output ranks and scores of various information retrieval models applied to the query video pair. The scoring models used in this work include term frequency inverse document frequency (TFIDF), probabilistic best match (BM25) [38] [21], and language models (LM) [55]. These models are applied to each of the text based fields: ASR [72], OCR, FileName, Title, Description, Subject, and Keywords. This category of features also includes the number of term matches, percentage of term matches, calculated term frequency (TF), and the calculated inverse document frequency

(IDF) for each text field.

The final category is video Features which are derived from the automated speech, metadata, or visual components of the video. The text-based features for ASR and metadata include term counts by field and identified person, location, and organization named entities. From the visual modalities, we derive both low-level and high-level image features.

Low-level features include edge [9] histograms, color histograms [8], and bag-of-visual words using SURF [47] keypoint features. The edge direction histogram provides a compact and computational efficient representation of the video. The descriptor divides the image into 16 sub-images, using a 4×4 grid, to allow for the calculation of localized edge distributions. The edge histogram consists of four directional edges and one non-direction edge for each of the sub-images and is represented by an 80-dimensional feature vector. Keypoint detectors [47] attempt to detect a small set of locally stable points and their surrounding region. Keypoints are clustered into a set of visual words to form the vocabulary for a bag-of-words. The size of the visual word vocabulary and the weight of each term is a parameter to the final representation.

Table 3.2: Semantic Concepts

Name	Definition
Airplane Flying	An airplane flying in the sky
Car	Shots of a car
Cityscape	View of a large urban setting, showing skylines and building tops.
Demonstration	One or more people protesting. May or may not have banners or signs
Road	Shots depicting a road

High-level features are the semantic concepts that we use to describe objects, events, and activities in videos. Table 3.2 shows example concepts and descriptions from the TRECVid [1] semantic indexing task. These concepts provide an approach to bridge the semantic gap between text descriptions and the low-level features of a video. A concept specific model is trained over the low-level features which can then be used to assign a confidence to a previously unseen video.

3.1.2 Pseudo positive examples

Training a machine learning algorithm over a feature vector, derived from a KIS query-video result, presents a challenge in the number of positive examples. Each query-video pair is assigned a label defining the relevance of the video to the given query. We formally define the query-video-label as a triple (q_i, v_j, l_k) where $l_k = (0, \ldots, 1)$, with 0 being not relevant and 1 being the most relevant. Consider the TRECVid KIS task, where a system is required to return a ranked list of the 100 top videos for each query. The output from this task results in one positive and 99 negative examples per query. This large class imbalance creates a challenge for any supervised learning algorithm.

Further inspection of the ranked result list, reveals that a number of negative examples are similar to the ground truth item in one or more modalities. Query 891 of the 2012 TRECVid KIS task states, "Find a video of yellow bus driving down winding road in front of building with flags on roof and driving past geysers". Consider the three example videos in Table 3.3. The first video, titled "100 th Anniv Old Faithful Inn Yellow Busses Ride the Old Road", is the ground truth video for query 891 and the two additional videos are clearly similar in title, metadata description, and video image. The identified similarities in multiple modalities show that while a single correct answer exists for a given KIS query, the problem could be generalized to one of graded relevance rather than simple binary classification. Identifying additional pseudo positive KIS examples helps to lessen the class imbalance problem and results in boosting similar videos higher in the ranked result list.

We propose a semi-supervised learning approach to KIS where the training set for a

Table 3.3: Similar Videos

Title: 100 th Anniv Old Faithful Inn Yellow Busses Ride the Old Road Desc: As a part of the Old Faithful Inn 100 th Anniversary Celebration yellow National Park busses...	
Title: Yellowstone Porcelain Basin Desc: ...warm spring morning to shoot video of the Porcelain Basin area within Norris Geyser Basin...	
Title: Yellowstone Snowloads Diminish and Lion Geyser Roars Desc: When the roads in Yellowstone are clear enough to safely allow cars...	

given query includes both the single truth example and a set of pseudo positive examples. The pseudo positive examples are identified by similarity to the truth video across all of its modalities. Each pseudo positive example is assigned a decreasing graded relevance based on the distance from the truth video. Using this approach we define the query-video-label as a triple (q_i, v_j, l_k) where $l_k = (0, \ldots, 4)$, with 0 being not relevant and 4 being the most relevant.

To identify pseudo positive examples we model the videos as nodes in a graph and the similarities as edges that estimate the importance of nodes in the graph relative to the root node. For a given KIS query q_i, we construct a directed graph $G_i = (V_i, E_i)$, where V_i is the set of video nodes in the query specific graph and E_i is the set of edges. We define the ordered pair (u, v) as the directed edge connecting video node u to video node v. The

query specific graph is iteratively constructed by initially selecting the truth video as the root node and performing a similarity search in the video collection using each modality. The result nodes from each iteration are used as search nodes for the next iteration. We define the iterative graph construction as follows: $\forall j = 1, \ldots, m$, video v_j is added to V_i and (u, v_i) is added to E_i if

$$F(u, v_j) > \alpha_t, \tag{3.2}$$

where m is the size of the video collection, and $F(u, v_j)$ is a modality similarity score between the root u and each video v_j in the collection. A modality specific threshold α_t is used to select the subset of videos. The α_t is empirically selected for each modality using a validation set. The final graph is modeled as a directed acyclic graph (DAG), and therefore does not contain loops or parallel edges. We use a K-Step Markov approach [69] to assign an importance measure $I(v|u)$ to each video node v in the graph with respect to the truth node u. The output is a ranked list of videos relative to the known item video.

3.1.3 Multimedia Rank Learning

Our approach to multimedia rank learning uses a framework based on gradient boosted regression trees [68] [73]. The gradient boosting framework uses a stage-wise approach to generate an ensemble of weak models, each a simple regression tree, that when combined produce a strong rank learning classifier. The algorithm uses regression trees to perform gradient descent in function space and can be trained to minimize a general differentiable loss function. The final ranking score is a linear combination of the output scores from each of the simple regression tree models.

The model is constructed using a training set of query-video pairs with known relevance labels. Each of the query-video pairs is mapped to an input feature space derived from queries, videos, and query-video dependent results. The regression tree algorithm learns a mapping of the input feature space to the known relevance label. The output on set a set of previously unseen query-video pairs is a ranked list of videos for the given query.

3.2 Experiments

Experiments are conducted using the known item queries and video repository from the TRECVid 2010-2012 [1] evaluation. This video collection consists of approximately 25,000 Internet Archive Videos distributed in MPEG-4/H.264 format and released under the Create Commons license. These videos total about 600 hours and have a duration between 10 seconds and 3.5 minutes.

Table 3.4: Example queries from TRECVid 2012 KIS

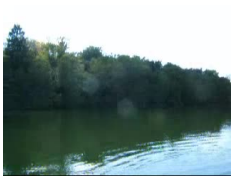
Query	Description	Video
893	Find the video of man speaking German with long hair and green jacket and soccer ball in a parking lot.	
909	Find the video of woman pouring black oil from milk carton.	
968	Find the video of three men, one with spiked white hair and black and red vest.	
1035	Find the video with a lake and its shores.	

Table 3.5: Example queries from TRECVid 2012 KIS

Query	**Description**	**Best Found in**
895	Find the video titled "Sunday Quickie" of a man who is wearing glasses and a blue shirt standing by the window and watching the rain outside and discussing his trip to Home Depot and Harveys Hamburger Kiosk.	Meta Title
1002	Find the video of man demonstrating use of children's laptop.	OCR
1051	Find the video of a shirtless boy playing with toy helicopter, gun and soldiers.	Meta Desc
1115	Find the video of a close-up face shot of a man wearing dark glasses and a white shirt who is giving a satire X-lawyer advertisement.	FileName
1167	Find the video titled "Welcome to Best Bible Study on Earth" which starts with a picture showing the mountains, lake, and sky and then a map of the United States where the narrator solicits you to go to their website.	ASR

Table 3.4 provides a sample of the queries and ground truth known item images from the repository. Queries are provided to the system as a text only description of both the audio and visual components of the video. The video repository includes the MPEG-4/H.264 video, the original collected author metadata, and speech to text. The system returns a ranked list of the top 100 videos most likely to match the queries. The system is evaluated using the mean inverted rank (MIR) for the known item queries and ground truth results of the TRECVid task. Table 3.5 shows an example of how the best field match for a query can be found in different metadata fields.

3.2.1 Analysis

Baseline experiments are conducted using a text only information retrieval approach [74], where the text modalities are merged. The metadata, ASR, and OCR from the repository

Table 3.6: TRECVid 2012 - Early Fusion MIR

Model	IR@1	IR@3	IR@5	IR@10	IR@100
TFIDF	0.25	0.30	0.30	0.31	0.32
BM25	0.29	0.31	0.33	0.33	0.34
LM	0.28	0.31	0.32	0.33	0.33

are used to generate a video document that can be indexed and retrieved using state-of-the-art retrieval algorithms. The results provide both a baseline comparison and a set of query-video features used by our rank learning algorithm. In this experiment, we merge all of the video text fields into a single document for indexing and retrieval. Both the queries and video documents are preprocessed for stop word removal, word stemming, and spell correction. The results in Table 3.6 show the MIR for the TRECVid 2012 task using three different retrieval models. The MIR is calculated at five different ranking points, starting at the top returned document and ending with document 100. Results show that the BM25 model produces the top MIR scores at each of the ranking points for the early fusion approach.

Table 3.7: TRECVid 2012 - Early Fusion Counts

Model	Ct@1	Ct@3	Ct@5	Ct@10	Ct@100
TFIDF	90	129	140	158	218
BM25	104	126	143	160	225
LM	102	129	143	154	221

Table 3.7 provides further analysis of our baseline early fusion models. This table shows a breakdown of the total documents found at each of the ranking points. The results show that BM25 outperformed the TFIDF model at rank 1 (MIR@1) by 14 KIS videos. It is also interesting to note that while 82 videos are identified after rank 5, they increase the final MIR by only .01.

Table 3.8: TRECVid 2012 - MIR by Field Type

Field	IR@1	IR@3	IR@5	IR@10	IR@100
ASR	0.09	0.10	0.10	0.10	0.10
OCR	0.07	0.09	0.09	0.09	0.09
File	0.10	0.13	0.14	0.14	0.15
Title	0.15	0.17	0.18	0.18	0.19
Desc	0.17	0.19	0.20	0.20	0.21
Keyword	0.00	0.00	0.00	0.00	0.01
Subject	0.08	0.10	0.11	0.11	0.11

The next set of experiments follow the information retrieval approach, but are performed on each of the text modalities. These experiments provided modality specific query-video features for the rank learning approach and help to identify the contribution of each of the text modalities. Tables 3.8 and 3.9 show the MIR and count found for the 361 queries, using a BM25 retrieval model for the seven text modalities of our video. The MIR and count are calculated at five different ranking points, starting at the top returned document and ending with document 100.

The metadata description provided the highest IR and count found. This modality is provided by the author and contains the least noise and most detailed description of

Table 3.9: TRECVid 2012 - Count by Field Type

Field	Ct@1	Ct@3	Ct@5	Ct@10	Ct@100
ASR	31	42	43	53	78
OCR	27	37	38	47	66
File	38	57	69	83	122
Title	54	74	85	94	142
Desc	62	79	91	109	162
Keyword	1	2	4	5	7
Subject	29	46	54	65	92

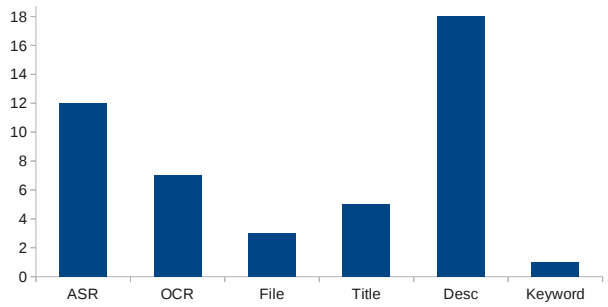


Figure 3.1: TRECVid 2012 - Unique Found by Field

the video. The results also show that the metadata Title and FileName identify a large number of relevant videos. Authors often include key terms in these fields that summarize the content of the video. ASR and OCR results suffer from noise generated during the translation from speech and video into text. However, Figure 3.1 shows that the ASR is comparable to the metadata Description for the number of unique ground truth videos identified.

3.2.2 Rank Learning

The Rank Learning experiments are evaluated over 100 runs using a 10-fold cross validation of the KIS query set, where each fold is divided into train, validate, and test. A feature vector is constructed for each query-video pair and consists of the query, query-video, and video features. The scores for the query-video features are derived from the models and fields described in the information retrieval analysis experiments. A keyframe is extracted every two seconds from each video in the collection to derive the set of visual features. OCR text for a video consists of the concatenated text extracted from each image frame. Edge, color, and local keypoint features are also extracted at every frame.

The gradient boosted regression trees used for ranking are trained using a cross-entropy cost function. To avoid model over-fitting, the number of trees is controlled by monitoring the prediction error on the validation set. Models were iteratively trained to a maximum of 1000 trees and results show that 100 tree provided good accuracy on the validation set. The maximum number of leaves per tree and learning rate were also used to control over-fitting by monitoring the validation set. The reported results use a maximum of 5 leaves per tree and learning rate of .05.

Table 3.10: TRECVid 2010-2012 KIS Results

Field	Description	2010	2011	2012
Early Fusion	IR model with all views merged	0.35	0.32	0.34
KIS Rank	Ranking model	0.38	0.33	0.36
SemiKIS Rank	Ranking model with pseudo positives	0.40*	0.36*	0.38*

Tables 3.10 provides the evaluation results from the TRECVid 2010-2012 KIS task. The Early Fusion approach uses a BM25 similarity model that combines the text representation from the audio, video, and metadata content. The KIS Rank model uses the derived feature space and gradient boosted regression trees to learn a model. This model does not use pseudo-positive examples and includes a single truth video and 99 negative videos for each training query. The results show that the KIS Rank approach provides an increase in MIR over the baseline Early Fusion approach.

The SemKIS Rank approach extends the KIS Rank model with pseudo positive examples, derived from the K-Step Markov graph ranking approach. The graph is iteratively constructed by selecting the KIS truth video as the root and performing a similarity search in the video collection using each of the seven text and the three visual modalities. SemKIS Rank allows max path lengths of 3 on the graph and identifies approximately 9 pseudo positive examples per query. Pseudo positive examples are assigned a graded relevance from 1 to 4 using the importance measure I assigned by the K-Step Markov approach. Our reported results assign relevance 4 for ground truth, 3 for $I >= .05$, 2 for $.05 > I >= .01$, and 1 for $.01 > I >= .001$. These values were determined using the validation set. These results show that the rank learning models are able to boost additional positive KIS examples higher in the ranked result list. A pairwise t-test over the SemiKIS Rank and KIS Rank results, show a statistically significant difference at the 95% confidence interval for all of the evaluation years.

Chapter 4: Multimedia Multiview Rankings

In the previous chapter, we introduced a semi-supervised rank learning algorithm for multimedia known item search. That approach uses pseudo-positive KIS examples to offset the class imbalance of having a single known item. In this chapter, we build on our previous approach by introducing, multimedia multiview ranking.

A multimedia collection consists of multiple modalities such as metadata, automated speech to text, optical character recognition, and visual features. Each modality provides a different view of a video and includes its own unique feature space. Queries to a multimedia repository are often verbose and includes phrases found in one or more views.

One of the challenges of the multiview problem is the mapping of a text query to the correct set of views. A baseline approach for video retrieval is to submit the full query to an index composed of the text representation of the videos. In this scenario the text modality views of the video, such as metadata, automated speech to text, and optical character recognition, are concatenated into a single view. The problem with this approach is that the influence of phrases which are more selective for a single view are diminished in the concatenated view. This is a similar problem to the query-drift [7] that is often found in the information retrieval community.

Our approach to the query mapping problem is to identify key phrases from the original query which align with a specific video view. Natural language processing techniques, such as named entity extraction and dependency tree parsing, are used to identify the key phrases from the original query. A supervised rank learning algorithm is used with a set of labeled queries to construct a model for identifying the correct view mappings. The model output on a set of previously unseen queries allows us to select N phrases from the ranked list and activate only those views relative to the given phrase.

A second challenge for the multiview problem is how to combine the result rankings from each of these unique views. Many of the video queries activate multiple views, which results in different view rankings for the same query-video pair. In addition, we exercise multiple information retrieval models, such as term frequency inverse document frequency (TFIDF), probabilistic best match (BM25) [38] [21], and language models (LM) [55], which produce additional ranking output.

To address this multiview challenge we introduce a rank learning approach to the video retrieval problem. Learning to rank is an approach used by the information retrieval community, where machine learning algorithms are used to create a ranking model for documents. The models are trained using a feature space derived from the queries, document, and query-document pairs. Our approach for multimedia rank learning follows a hierarchical model, where the output of view specific models are combined into a final multiview ranking. The hierarchical approach allows the initial view specific models to focus on their own unique feature space before attempting to merge results. Each view includes a unique description of the feature space which may be difficult to capture in a single ranking model.

Our contributions to the web search and data mining community include the following:

1. We model multimedia retrieval as a multiview problem. In the multiview model each of the modalities, such as metadata, automated speech to text, optical character recognition, and image, are treated as a unique view of the system.

2. We use natural language processing techniques to identify view specific phrases of a query and then output a ranked mapping of the phrases into their respective views. This approach allows us to identify and activate only those views of the video which are applicable to the given query.

3. We model the individual feature space for each multimedia view and create a view specific model using gradient boosted regression trees.

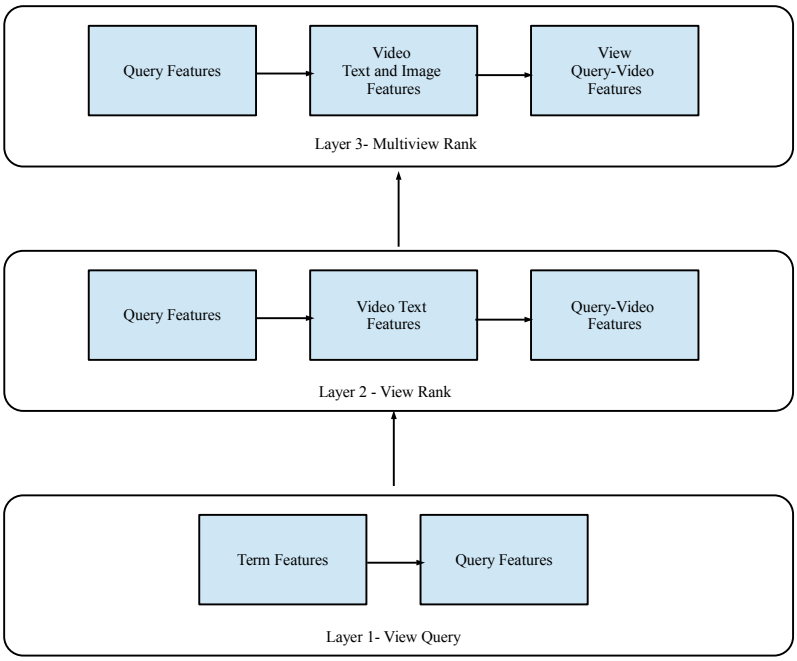


Figure 4.1: $Multi^2Rank$

4.1 Our Approach

We propose a hierarchical multimedia multiview rank learning model, called $Multi^2Rank$, to address the challenges of this unique multimedia retrieval problem. Figure 4.1 provides an overview of the model. The first layer uses natural language processing techniques to identify view specific phrases within a query. The second layer models the individual feature space for each view and creates a view specific ranking model. Output from each of the view ranking models is passed to the final ranking layer of the hierarchy, where a multiview model combines each view to create the final video ranking.

4.1.1 View Query

Queries to a video retrieval system often take the form of a text only description and return a ranked result set from an index which concatenates the videos multiple views. The text only query includes multiple phrases which identify features of a specific view. This multiview problem presents a challenge in mapping these phrases into the correct view feature space. The first layer of our model uses natural language processing techniques to identify view specific phrases and output a ranked mapping of the phrases into their respective views.

Table 4.1: Example Known Item Search Queries

Query ID	**Description**
002	Find the video of an Sega video game advertisement that shows tanks and futuristic walking weapons called Hounds.
074	Find the video of George Bush with red tie and men with microphone.
116	Find the video of a man who has his arm resting on a parking meter, then he is looking through binoculars and waving and there are sailboats moving on the water.
171	Find video of a Coast Guard advertisement showing a man in red wet suit, a woman in a blue uniform and men in dress whites and depicts a rescue in a stormy sea featuring a man in sea waving and persons jumping from helcopter, an armed ship boarding and an oil spill operation.
229	Find the black and white video titled "Powers Case" which shows President Eisenhower before microphones giving a talk about the Powers Spy Case and his Open Skies proposal.

A user query to a video retrieval system is often overly descriptive and includes phrases from multiple views. Table 4.1 provides an example set of video queries from the TRECVid [1] evaluation. This problem is compounded for known item queries where the user provides additional descriptive phrases in order to identify the single correct video. Consider the following query from the TRECVid 2010 test set, "Find the video of the film with Robert Hoffman and Briana showing scenes of wild and feverish street dancing and classes at the Maryland School for the Arts". A baseline approach to this problem is to create a text representation of the video using speech to text, optical character recognition, semantic concept identification, and metadata such as filename, title, description, subject, and keywords. This query can then be submitted to a retrieval system using a probabilistic scoring model and indexed over the video text representation. This simple baseline approach returns the relevant document at rank 22, which is well outside the first page of results returned by a typical browser.

We believe that a better approach is to identify key phrases from the original query which align with a specific video view. The view specific query is then submitted to a retrieval system indexed on the corresponding view. In this scenario, the example query is broken into the two metadata specific phrases, "robert hoffman briana" and "maryland school", which return the known item video at rank 1.

To select candidate terms, a probabilistic natural language parser is used to identify grammatical relationships in the video query. Terms from noun phrases are captured to form the basic candidates for phrase selection. Research from the information retrieval community has shown that these types of phrases containing nouns and verbs are generally more informative in a query [75] [76].

The next step is to construct a feature space for the candidate terms in order to train our ranking model. The view query feature space is defined in Table 4.2 and consists of term and query features. Query features provide information about the full query such as its length or its bag of words. Term features describe the current ranking term and include attributes such named entity, part of speech, and edge parse. A named entity parser [71] is used to

Table 4.2: View Query Features

Type	**Feature**
Term	Is All Upper Case
	Has upper case first letter
	Is All Numbers
	Term is enclosed in a quote
	Character N-Gram
	Named Entity (NE)
	Part of Speech (POS)
	Is a NNP
	View term frequency
	View document frequency
	Incoming edges of the parse
	Outgoing edges of the parse
	Term
	Prefix
	Suffix
	Is Stop Word
	Length of the term
	Starts with Upper Case
Query	First Term
	Last Term
	Bag of POS
	Bag of Words
	Topic Length

identify entities such as person, location, or organization. An example named entity parse is shown in Figure 4.2 for a TRECVid query. Named entities identify informative phrases that are often found in video metadata such as filenames and titles. These phrases can also appear in the visual view of a video as a title or subtext. The part of speech [77] for each query term identifies its lexical category and is also used to provide context for the candidate phrases. The dependency tree [78] provides a count of the incoming and outgoing edges to each of the candidate terms. Figure 4.3 shows a partial dependency parse from

```
TRECVid Topic 80
Find the video of Marion-PERSON Carroll-PERSON sitting in front
of computer and the sound is his request for election to
Admissions-ORGANIZATION Committee-ORGANIZATION at Xavier-LOCATION.
```

Figure 4.2: Named Entity

```
(NP
 (NP(DT the)(NNP Maryland)
    (NNP School))
 (PP (IN for)
  (NP (DT the)(NNPS Arts)))))))
```

Figure 4.3: Query Parse

a TRECVid query. Additional features such as quotes and capitalization help to identify titles and known locations.

To train a view query model, we map each term into its feature space and assign a graded relevance label of 0 to 4. The label represents the degree of relevance for the current term to the given view. We use gradient boosted regression trees to train the query ranking model. This is the same ranking algorithm used in all three layers of our model.

4.1.2 View Rank

The view rank layer models the individual feature space for each view and creates a view specific ranking model. Figure 4.4 shows an overview of our view ranking approach. To train the model, we first index each of the individual video views. A view specific query, generated from the layer one model, is then used to create query-view rankings. Next, we construct the model feature space using the query, video, and query-video features. The

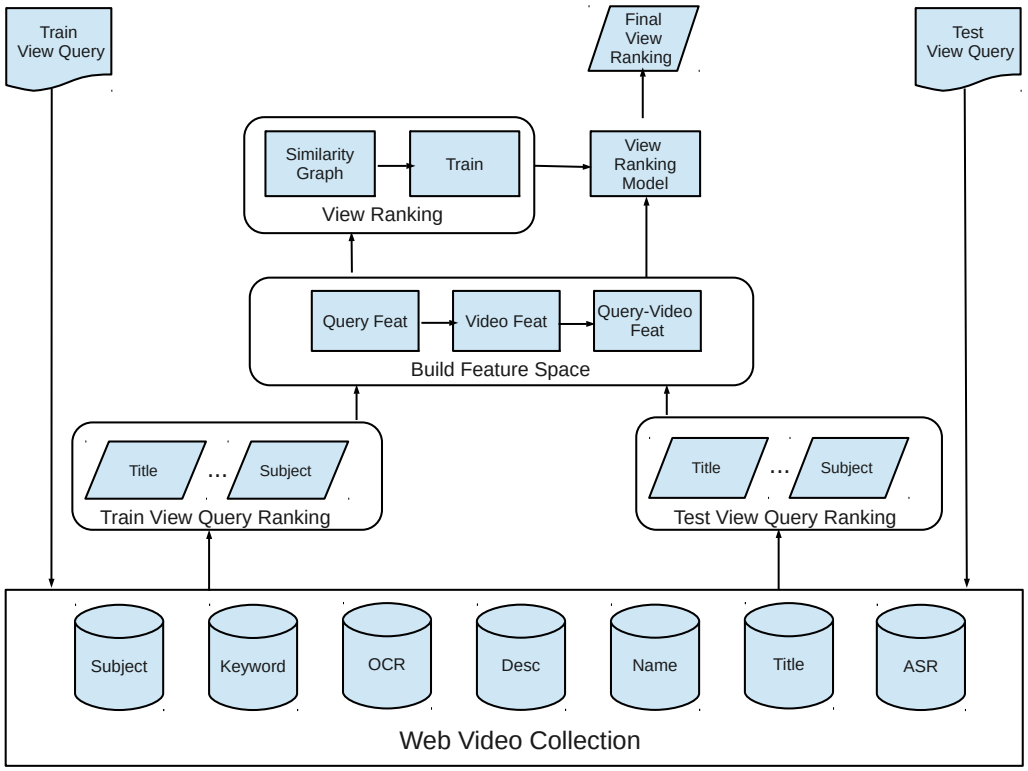


Figure 4.4: Layer 2 - View Rank Model

model feature vector is labeled with a ground truth relevance value and passed to our tree-based ranking algorithm to create a view specific ranking model. Each view test query follows a similar path of phrase ranking, feature space construction, and view ranking.

Each video is comprised of speech to text, image frames, semantic concept classifiers, and metadata consisting of filename, title, subject, keywords, and description. Table 4.3 shows an example of the metadata associated with a video from the TRECVid evaluation. The view rank feature space is defined in Table 4.4 and consists of query, video, and query-video features. The query features are derived from the view query and provide information

about its length, bag of words, and NLP features. The video features are derived from the current video view and include term count and NLP features. The query-video features are derived from the output ranks and scores of various information retrieval models.

Each training query-video pair is assigned a label defining the relevance of the video to the given query and creates the triple (q_i, v_j, l_k). To identify additional relevant training videos we follow our semi-supervised approach using the K-Step Markov [69]. This allows us to use a set of relevance labels with 0 being not relevant and 4 being the most relevant. A known item is labeled with relevance of 4 and those videos similar to the truth are provided a graded relevance based on their similarity distance to the known item. For the supervised rank learning model we follow our Gradient Boosted Regression Trees approach and iteratively create an ensemble of weak regression tree models.

4.1.3 Multiview Rank

The final layer of our model combines each of the view rankings to create a final multimedia multiview ranking. The multiview feature space is defined in Table 4.5 and is constructed from the full query, video, and query-video features. The query-video features are derived from the output ranks and scores of the layer two view rank models. The feature space for the visual view of the video includes a visual bag of words and semantic concept set. Table 4.6 shows a set of sample image frames from TRECVid evaluation video. A visual bag of words is defined over a set of Speeded-up Robust Features (SURF) [47], which have been clustered into a vocabulary of visual words. SURF is a local interest point detector and descriptor which identifies distinctive locations in an image. These interest points and descriptors represent corners, blobs, and t-junctions and have proven effective for object recognition. Clustering of the descriptors on a training set allows us to create a vocabulary of interest points which can be used for image classification and search. Semantic concepts are high level features which describe objects and events within image data. These concepts help bridge the understanding gap that exists between text and the low level features of an image. Examples of semantic concepts include: desert, flowers, girl, hospital, and person.

We use a set of 130 semantic concepts identified in the work of [34] [79] for the TRECVid 2010 web video data. The final concepts classifier [34] uses an SVM to fuse six individual classifiers trained on local keypoint features and global image features, such as edge, color, and texture. The use of these features allows the tree based algorithm to identify ranking patterns within sub groups of queries containing similar semantic concepts. The K-Step Markov approach is used to identify similar videos to the known item and a graded relevance score is assigned. Gradient Boosted Regression Tree are used to train the top level ranking model, which combine the ranks from each of the multiviews.

4.2 Experiments

We evaluate $Multi^2Rank$ using the benchmark data and truth set from the TRECVid 2010-2012 [1] Known Item Search task. Experiments are conducted using a set of approximately 25,000 internet videos, licensed through Creative Commons. The videos total approximately 600 hours total and range in duration from 10 seconds to 3.5 minutes. Content covers a wide range of topics including short documentaries, home videos, and commercials. Videos also includes metadata in the form of filename, title, subject, keywords, and description provided by the video donor.

The Known Item Search task models the scenario of a user attempting to find a specific video in a large internet collection. The assumption is that the user has seen the video before and uses a text only query to describe what he remembers about the video. Query creators were asked to view a random video and then create a text description which is specific enough to identify that particular video.

The evaluation set includes 1000 queries, approximately 300 per year, drawn at random from the internet video archive. The task goal is for the system to return a ranked list of the top 100 videos most likely to match the description. Mean inverted rank is used to measure the performance of a known item system.

4.2.1 Analysis

A set of baseline experiments provide us with a benchmark comparison using a traditional text-based information retrieval model. In this initial experiment, we index each view of the video collection and then use the full text query to obtain a ranked list of videos. Results using a BM25 [38] information retrieval model are show in Table 4.7. The video is segmented into the 7 views: OCR, ASR, fileName, title, description, keywords, and subject. The baseline results show that the title and description provide the top MIR results of .20 and .24. The title score is somewhat surprising given that a title is normally a short phrase versus the description which often includes multiple sentences. Table 4.8 identifies the number of known items found only in a given view. It is interesting to note that while the OCR view is one of the worst MIR views, it is the top view for unique known items found.

4.2.2 Ranking Results

To prove that our hierarchical ranking model improves over the baseline approach, we conduct a series of experiments and show results at each layer of the model. Experiments are evaluated over 100 runs using a 10-fold cross validation of the queries from the 2010-2012 TRECVid evaluation and scores are reported using the mean inverted rank. As described in the approach section, each layer of the model uses a ranking model based on gradient boosted regression trees with a cross-entropy cost function.

Our first set of experiments evaluate the use of the view query model. This approach builds a supervised model which identifies view specific phrases, which are used to query the individual indexed views. The model allows us to filter noisy queries to identify a small set of view phrases. The training and test sets are mapped into the query feature space defined in Table 4.2. To identify labels for the supervised ranking model, we select matching terms and scores for a query-video pair using a BM25 information retrieval model. The candidate terms are assigned a relevance label from 1 to 4 based on their matching score. The tree based models are iteratively trained to select the best parameters using a validation set. We avoid over-fitting by controlling the learning rate and the maximum number of leaves.

Our results are shown with 100 trees, with a maximum of 5 leaves, and a learning rate of .05.

The results in Table 4.9 show the multiview activation results from the TRECVid 2010 KIS evaluation. The title shows a MIR improvement from 0.20 to 0.25 and the description increases from 0.24 to 0.28. However, not all of the views improve using the query view models. Both the OCR and keyword views decrease in MIR compared with the baseline information retrieval approach. This decrease occurs because the views are only able to identify a relatively small number of known items. As a result, they provide relatively few training examples for our model.

The second set of experiments evaluate the use of the view ranking model. A model is trained for each view in its own unique feature space, which should improve the MIR for that view. The training and test sets are mapped into the view ranking feature space defined in Table 4.4. The dependent features for this feature space are derived from the output ranking and scores of the view specific query model. The K-Step Markov approach is used to identify labels for the training of the ranking model. A maximum path length of 3 is used to derive the importance measure I on the graph and identify candidate videos. The candidate videos are assigned a relevance label as follows: 4 for ground truth, 3 for $I >= .05$, 2 for $.05 > I >= .01$, and 1 for $.01 > I >= .001$. The results in Table 4.10, from the TRECVid 2010 KIS evaluation, show that the view ranking model improves the ranks for every view, when compared with the view query results. The title and description continue to be our top performing views with MIR of 0.29 and 0.33.

The last set of experiments provide the overall results for the TRECVid 2010-2012 KIS evaluation. We compare our $Multi^2Rank$ model with Early Fusion, KIS Ranking, Semi-Supervised KIS Ranking, and Late Fusion [4]. The results in Table 4.11 show that the multiview approach improves over our initial KIS Ranking and Semi-Supervised models. Our $Multi^2Rank$ results also improve over the Late Fusion model for all three years of the TRECVid KIS evaluations. We see a .03 MIR increase for TRECVid 2010, .04 in TRECVid 2011, and .03 in TRECVid 2012. A pairwise t-test over the $Multi^2Rank$ and Late Fusion

results, show a statistically significant difference at the 95% confidence interval for all of the evaluation years.

Table 4.3: Example Known Item

Query ID	**Description**
Topic 073	Find the video of many engineers in hard hats descending into the tunnels of a subway system in Berlin.
File Name	Mayda 3000 Construction Tour Of U 55916 Mayda 3000 Construction Tour Of U 55916
Meta title	Construction tour of U55
Meta desc	Engineers and railway geeks rejoice Tom went on a tour of the construction site of the new subway line U55 at Brandenburger Tor Braving the noise and dust we can take a first look at the extension of the U5 and the new subway station on Pariser Platz.
OCR	Cahstruction tol 11 r h Jnev subway LUJ VIM Ash Cfmstrucition 1t9 w6ffhe Lnevvx subway M Av DA 3000 BL05 s Po T co M Cahstrucltion dfthnew subway Cahstrucltion dfthnew subway
Speech	All right he corresponded Williamson on the case here at you 55 subway in downtown Berlin Among the go underground and get the dirt on this story I m about and 30 other engineers ... construction site and finally to left You re going to see a cement factory pushing cement down unbelievably loud noise into the room to be used down the tunnel

Table 4.4: View Rank Features

Type	**Feature**
Query	First Term
	Last Term
	Bag of POS
	Bag of NE
	Bag of Words
	Topic Length
Query-Video	TFIDF Weight Model
	BM25 Probabilistic Model
	Language Model
	Percentage of Term Match
	Inverse Document Freq of Match
	Term Freq of Match
Video	Bag of POS
	Bag of NE
	Bag of Words
	Length

Table 4.5: Multiview Rank Features

Type	**Feature**
Query	First Term
	Last Term
	Bag of POS
	Bag of NE
	Bag of Words
	Topic Length
Query-Video	TFIDF Weight Model
	BM25 Probabilistic Model
	Language Model
	Percentage of Term Match
	Inverse Document Freq of Match
	Term Freq of Match
Video	Semantic concepts
	Visual Bag-of-words
	Bag of POS
	Bag of NE
	Bag of Words
	Topic Length

Table 4.6: Example known item image frames

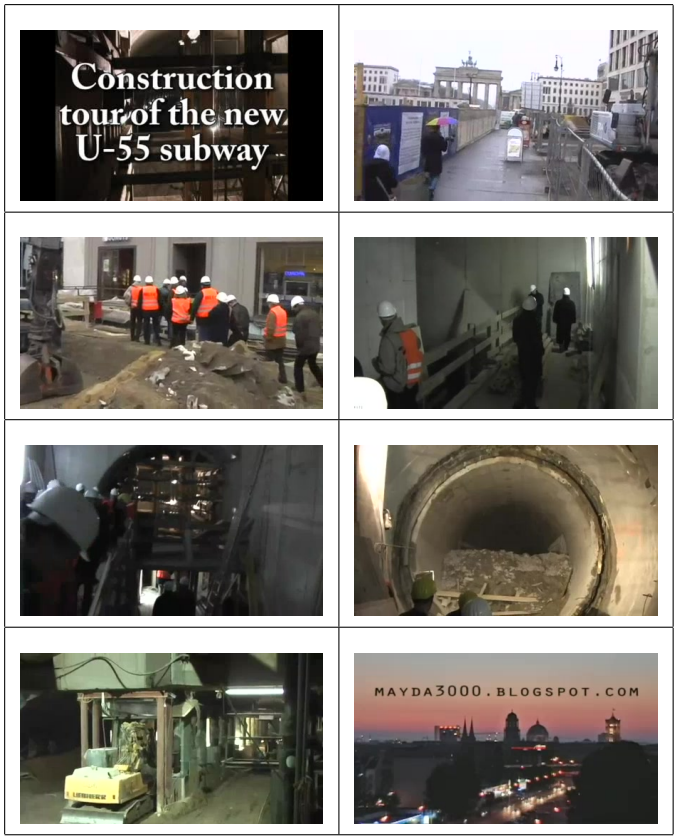


Table 4.7: TRECVid 2010 KIS Results by View

Field	MIR@1	MIR@3	MIR@5	MIR@10	MIR@100
ASR	0.07	0.08	0.08	0.08	0.08
OCR	0.04	0.05	0.05	0.05	0.05
File	0.13	0.15	0.15	0.16	0.16
Title	0.16	0.18	0.19	0.19	0.20
Desc	0.20	0.22	0.23	0.23	0.24
Keyword	0.02	0.02	0.02	0.02	0.02
Subject	0.07	0.09	0.09	0.09	0.09

Table 4.8: TRECVid 2010 Unique Found by View

Field	Count
ASR	0
OCR	5
File	3
Title	4
Desc	3
Keyword	1
Subject	4

Table 4.9: TRECVid 2010 View Specific Query Results

Field	IR@1	IR@3	IR@5	IR@10	IR@100
ASR	0.10	0.11	0.11	0.11	0.12
OCR	0.03	0.03	0.04	0.04	0.04
File	0.20	0.22	0.22	0.22	0.22
Title	0.20	0.23	0.24	0.25	0.25
Desc	0.23	0.26	0.26	0.27	0.28
Keyword	0.01	0.01	0.01	0.01	0.01
Subject	0.10	0.12	0.12	0.13	0.13

Table 4.10: TRECVid 2010 View Ranking

Field	MIR@1	MIR@3	MIR@5	MIR@10	MIR@100
ASR	0.13	0.14	0.15	0.15	0.15
OCR	0.03	0.04	0.04	0.04	0.04
File	0.20	0.23	0.24	0.24	0.24
Title	0.24	0.28	0.28	0.29	0.29
Desc	0.28	0.31	0.32	0.32	0.33
Keyword	0.01	0.01	0.01	0.01	0.01
Subject	0.12	0.13	0.13	0.14	0.14

Table 4.11: TRECVid 2010-2012 KIS Results

Field	Description	2010	2011	2012
Early Fusion	IR model with all views merged	0.35	0.32	0.34
KIS Rank	Ranking model	0.38	0.33	0.36
SemiKIS Rank	Ranking model with pseudo positives	0.40	0.36	0.38
Late Fusion	Query to Modality Classification	0.41	0.35	0.39
$Multi^2Rank$	Multimedia Multiview Rank	0.44*	0.39*	0.42*

Chapter 5: Query Concept Ranking

The previous chapter introduced a multimedia multiview ranking approach for KIS, where each unique modality is represented as a view in the ranking feature space. Over the next two chapters we introduce a query-by-concept approach for KIS, where modality views are mapped into a semantic feature space for retrieval.

Semantic concepts provide a common representation for searching a multimedia repository. A query-to-concept approach for multimedia retrieval attempts to map the text query and multimedia modalities, such as image, speech, and metadata, into a common feature representation.

Query-to-concept mapping is a challenging task, since the text query often provides little context for identifying representative semantic concepts. Concept drift can occur when a verbose query results in the selection of semantic concepts unrelated to the primary query topic. In addition the semantic concept vocabulary size is often limited, and a query-to-concept mapping must include a sufficient number of semantic concepts to identify and return a ranked list from the repository. This problem is compounded in a Multimedia Known Item Search (KIS), where a searcher attempts to find a previously viewed video in the repository. This type of query requires an expanded semantic concept vocabulary set to uniquely identify the single video.

To overcome the challenges of query-to-concept mapping for KIS we propose a concept ranking model, called ConRank. First, we introduce natural language processing techniques to identify key phrases within the query for effective mapping into a set of semantic concepts. Next, we model the query-to-concept mapping as a concept ranking problem, where a ranked list of semantic concepts is identified for each query. We evaluate our approach using a set of KIS queries and truth semantic concept annotations from the TRECVid evaluation.

Our contributions to the multimedia retrieval community include the following:

1. We use natural language processing techniques to extract key phrases from a KIS text query in order to reduce semantic concept drift.

2. We construct a ranking feature space derived from queries, semantic concepts, and partial rankings.

3. Finally, we introduce a concept ranking model to map a KIS text query into a semantic query.

5.1 Our Approach

We propose a concept ranking algorithm, called ConRank, to map a text query into a set of semantic concepts. Given a set of n text queries $Q = (q_1, \ldots, q_n)$ and a set of d semantic concepts $C = (c_1, \ldots, c_d)$, we formally define the concept ranking problem $R(q_i, C) = (r^i_{c_1}, \ldots, r^i_{c_d})$, where $r^i_{c_j}$ is the rank of concept c_j for query q_i.

Figure 5.1 provides a graphical description of this approach. Given a text query, natural language processing techniques are used to identify key phrases within the text query. These phrases are used to generate a set of partial rankings using a similarity comparison to the semantic concept set and the annotated multimedia repository. A concept ranking model is then used to combine the rich feature space of our partial rankings and provide the final ranked list of semantic concepts. The semantic query can then be used for KIS retrieval and ranking in a common feature space.

5.1.1 Phrase Selection

Mapping a text query into a set of semantic concepts requires a similarity comparison between the query description and the semantic concept set. However, the verbose nature of the query description often leads to a concept drift, similar to the query-drift [7] found in information retrieval. Consider the following query taken from the TRECVid 2012 KIS task [1]:

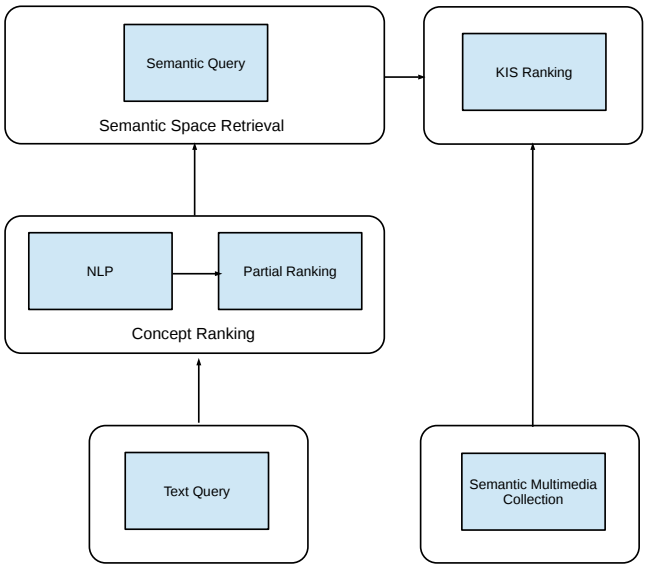


Figure 5.1: Concept Ranking Model

```
Find the video by CarDataVideo with music background that displays
the electric car, the Tesla Electric Roadster driving in several scenes
with words such as design and performance appearing on the screen.
```

The query consists of 32 terms that describe music, objects, organizations, and screen-display text. The known item video and metadata for this query is found in Table 5.1. Concept drift occurs when a verbose query results in the selection of semantic concepts unrelated to the primary query topic.

To overcome the concept drift problem for query-to-concept mapping, our approach uses natural language processing techniques, such as named entity recognition and parsing to identify key terms and phrases within the query. Named Entity Recognition (NER) [70] is

Table 5.1: Video with metadata

Video Shots	
File Name	Cardatavideo All Electric Tesla Roadster
OCR	Tesla Electric Roadster pmemea Tesla Electric Roadster presented Carnatavideo
Title	All Electric Tesla Performance Roadster
Desc	View the new All Electric Tesla High Performance 2 seat roadster.

an information extraction technique for detecting categories of text such as person, location, and organization. These named categories provide a key view for mapping our text queries into a set of semantic concepts. Two named entities extracted from our example KIS query are as follows:

```
CarDataVideo, Tesla Electric Roadster
```

The named entity view restricts the candidate set of semantic concepts to those related to the entity categories of person, location, or organization.

Parsing is a technique for identifying and extracting informative phrases within text [75] [76]. Our approach uses a probabilistic parser [80] to extract noun and verb phrases from the text query. Two phrases extracted from our example KIS query are as follows:

```
music background, the electric car
```

The use of phrases helps to overcome the concept drift problem and reduces the candidate semantic concept set. The phrase, "the electric car", reduces the candidate semantic concept set to those semantic concepts related to "Cars". The second phrase, "music background", helps to identify semantic concepts related to sound or music.

5.1.2 Concept Ranking

Table 5.2: Ranking Features

Type	**Feature**
Query	Phrase Count NER Count NER Histogram Length
Semantic Concept	Co-occur Histogram Concept Freq Concept VidFreq Length
Partial Ranking	NER Phrase

We define a ranking feature space for each query-concept pair, which is defined over the query, semantic concept, and partial rankings. The ranking feature space, found in Table 5.2, is formally defined as $(v^{q_i}, v^{c_t}, r^{c_t}_{q_i})$, where v^{q_i} are query features, v^{c_t} are semantic concept features, and $r^{c_t}_{q_i}$ are the partial ranking features.

Query features are derived from the original text query and the extracted query phrases. These features include a NER histogram, NER count, phrase count, and query length. This

type of feature has been used by the information retrieval community [63] to learn rankings between a query-document pair. In our model, the query feature is used to help identify sub-spaces with the training data for query-concept pair ranking.

Semantic concept features are those derived from the semantic concept set. Each semantic concept identifies a name, text descriptions, and a set of related semantic concepts. The semantic concept also includes aggregated features from the annotated multimedia repository, such as concept frequency, concept co-occurrence, named entity list, and key phrases. Our ranking feature space includes a semantic co-occurrence histogram, concept frequency, concept video frequency, and length. The semantic concept features are used by the learning algorithm, in combination with the query features, to identify sub-spaces with the training data.

Partial rankings provide the final category of features defined in our concept ranking algorithm. This set of features is derived from the output ranking of information retrieval models defined over a query-concept pair. These partial rankings are created using both probabilistic [21] and language [55] models.

ConRank uses our rank learning approach to create the final ranked set of semantic concepts for a given query. The model is constructed using a training set of query-concept pairs with known relevance labels. Each of the query-concept pairs is mapped to an input feature space derived from queries, concepts, and query-concept dependent results. The ranking model learns a mapping of the input feature space to the known relevance label. The output on set a set of previously unseen query-concept pairs is a ranked list of concepts for the given query.

5.2 Experiments

We evaluate our ConRank algorithm using the KIS query set from the TRECVid 2012 evaluation [1]. The 2012 evaluation includes approximately 350 KIS queries derived from a repository of about 8000 internet videos. Table 5.3 provides an example of KIS queries from the TRECVid 2012 test set. Our task is to perform a mapping of the text queries into

Table 5.3: Known Item Search Queries

Query	**Concepts**	**Video Shot**
Find the video of a helicopter taking off with three people and flying over land, water, and animals.	Airplane Person Daytime-Outdoor Sky Fields Forest	
Find the video with the left side of a rainbow. It is dark outside and there are mountains in the background to the right and shrubbery in the foreground.	Landscape Mountain Outdoor Plant Sky Hill Vegetation Trees	

a set of semantic concepts. The truth semantic concept set is derived from a subset of the 500 semantic concepts used during the TRECVid collaborative annotation [44] [45] effort. To evaluate the query-to-concept mapping, we measure precision, recall, and F-Score over the query test set.

The KIS task models a scenario where a searcher would like to find a previously viewed video from a large repository. The KIS query is formulated as a text only description of what the searcher remembers about the video. The queries are drawn from the IACC.1 internet video repository. The Creative Common license videos are between 10 seconds and 3.5 minutes in duration. The domain ranges from home videos to professional advertisements in MPEG-4/H.264 format and includes metadata provides by the content donor.

Our baseline experiment uses a probabilistic [21] retrieval model to compare the text query with the semantic concept descriptions. We select the top 7 semantic concepts from the resulting ranked list and compare them with the truth annotation set. The baseline

Table 5.4: Results

Model	**Precision**	**Recall**	**FScore**
Baseline	0.09	0.05	0.07
NER	0.30	0.09	0.14
Phrase	0.37	0.39	0.38
ConRank	0.42	0.38	0.40

results in Table 5.4 show that this approach suffers from concept drift and results in both low precision and recall.

The second experiment uses the named entities extracted from both the query and semantic concept set to generate a partial ranking. This approach also uses a probabilistic [21] retrieval model to generate a ranked list of candidate semantic concepts. The results in Table 5.4 show that this approach produces high precision results, but low recall due to the small number of entities identified.

The next experiment uses a probabilistic [21] retrieval model with the phrases extracted from both the query and semantic concept set. The results show that key phrase extraction mitigates concept drift and increases both precision and recall.

The final experiment shows the results from our ConRank approach. The supervised model is trained with the truth query-concept pairs from the KIS TRECVid 2010-2011 evaluation [1]. The truth set includes approximately 600 queries and truth annotations derived from the collaborative annotation [44] [45] effort. The final results show that ConRank is able to take advantage of the rich ranking features space to improve query-concept mapping. One example of the ConRank advantage can be found in KIS query 1205, concerning a commercial video for an animation premier. The NER partial ranking was able to identify the semantic concept, Commercial Advertisement. However, the phrase partial ranking was

not able to identify any of the truth semantic concepts. The ConRank model was able to learn the contributions from these partial rankings and identified the correct semantic concept.

Chapter 6: Semantic Rank Learning

In this chapter, we continue our discussion of a query-by-concept approach for KIS. The focus of the previous chapter was on the problem of mapping a KIS text query into a set of semantic concepts. In this chapter, we focus on the semantic concept mapping from the perspective of the multimedia repository.

One of the goals of multimedia retrieval is to find a common representation between a text user query and the multiple modalities of multimedia data. The use of semantic concepts is one approach to overcoming this representation gap. Semantic concepts are text labels, such as Snow or Person, that can be assigned to multimedia data such as frames of a video. These concepts provides a high level feature representation of the multimedia data. Table 6.1 provides an example of the semantic concepts used in the TRECVid evaluation [1].

The semantic concept representation has proven to be an effective approach for classification and indexing tasks [1]. Research in the retrieval community has included semantic concept for the search task, but generally as an additional feature to the metadata that continues to dominate retrieval performance.

One of the problems with using semantic concepts for retrieval is the size of the available concept vocabulary. Obtaining labeled data, extracting multimodal features, and training classifiers for each semantic concept is a difficult and time consuming task. Retrieval over an internet size repository requires a large number of concepts to accurately reflect the content and identify an item at query time.

The reliability of the labels produced by concept classifiers, presents a second problem for multimedia retrieval. Many of these models suffer from low precision due to a lack of training data and visual feature representation.

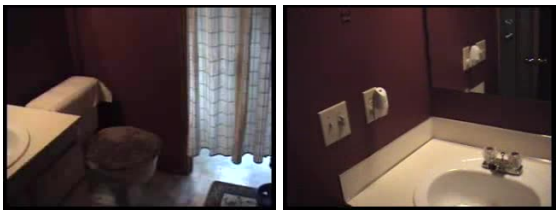
In order to overcome the vocabulary problem, we model semantic multimedia retrieval as a rank learning problem [35]. Our semantic rank learning model, called SemRank, is a supervised learning approach that builds on the traditional unsupervised information retrieval models by including features from both the queries and videos. Our approach derives a semantic feature space and trains a semantic ranking model using gradient boosted regression trees [58]. To improve the quality of concept labels, we propose a semantic fusion model to combine the labels from many weak classifiers. These classifiers represent different systems trained on varying data and modalities for a given concept.

Table 6.1: Semantic Concepts

Name	**Definition**
Airplane	Shots of an airplane.
Person	Shots depicting a person (the face may or may not be visible).
Prisoner	Shots depicting a captive person, e.g., imprisoned, behind bars, in jail or in handcuffs, etc.
Snow	Snow falling or already accumulated on the ground.
Suburban	Shots depicting an urban or suburban setting.

We evaluate our semantic retrieval approach using a set of KIS queries over a large internet video collection. In the KIS retrieval scenario, the user creates a query based on what he remembers and the system returns a ranked list of videos most likely to match

Table 6.2: Known Item Search Queries

Query ID	**Description**	**Video Shots**
912	Find the video of bathroom with brown walls, checked curtains and picture of camel on wall.	
961	Find the video set in a book shop. A dark-haired woman reads aloud from a book, which has a red cover with a heart on it.	
1042	Find the video showing a Komodo Dragon and a Lionfish.	

the request. Table 6.2 provides an example of KIS queries from the TRECVid evaluation [1]. This unique multimedia retrieval problem provides a good test environment for our semantic retrieval approach, since we are attempting to identify a single correct answer from a large collection.

Our contributions to the semantic computing community include the following:

1. We introduce a query-by-concept approach for multimedia Known Item Search (KIS)

2. We construct a semantic fusion model, to fuse labels from noisy concept classifiers and improve retrieval performance.

3. Finally, we propose a semantic rank learning model, called SemRank, and show improved ranking over traditional information retrieval models.

6.1 Our Approach

We propose a rank learning approach [35] to semantic concept based multimedia retrieval. Given a set of weak concept labels from multiple classifiers, a semantic fusion approach is used to derive a final semantic labeling. Queries to the retrieval system are provided in the form of a concept set and matched to the video semantic labels using a probabilistic model [21]. Rank learning improves on traditional information retrieval models by considering features of both the video and query, in addition to their feature similarity.

The semantic concept space is defined as $C = (c_1, \ldots, c_d)$, where d is the dimensionality of the concept space. Given a query set $Q = (q_1, \ldots, q_n)$ of size n, a mapping to the semantic concept space is defined as $F_q(q_i, C) = (q^i_{c_1}, \ldots, q^i_{c_d})$, where $q^i_{c_x}$ is the score for concept c_x in query q_i. A similar mapping is defined for the video repository $V = (v_1, \ldots, v_m)$ of size m, such that $F_v(v_y, C) = (v^y_{c_1}, \ldots, v^y_{c_d})$, where $v^y_{c_x}$ is a score representing concept c_x in video v_y. The multimedia retrieval problem is then defined as a ranking $R(q_i, V) = (r^i_{v_1}, \ldots, r^i_{v_m})$, where V is the video repository, q_i is a video query, and $r^i_{v_t}$ is the rank of video v_t for the given query.

Figure 6.1 provides an overview of the SemRank retrieval model. The video repository consists of a collection of internet videos and their associated metadata [1]. The metadata can include multiple modalities such as speech to text, optical character recognition, image features, and descriptive text provided by the video author. The repository is used as input to systems which provide semantic concept classifiers. Each system is trained using one or more modalities using both in-domain and out-of-domain data [1]. Given a video from the repository, a set of concept labels and scores are provided from each of the systems. Next, we train a semantic fusion classifier using all of the system output labels and derive a final concept set for the semantic repository.

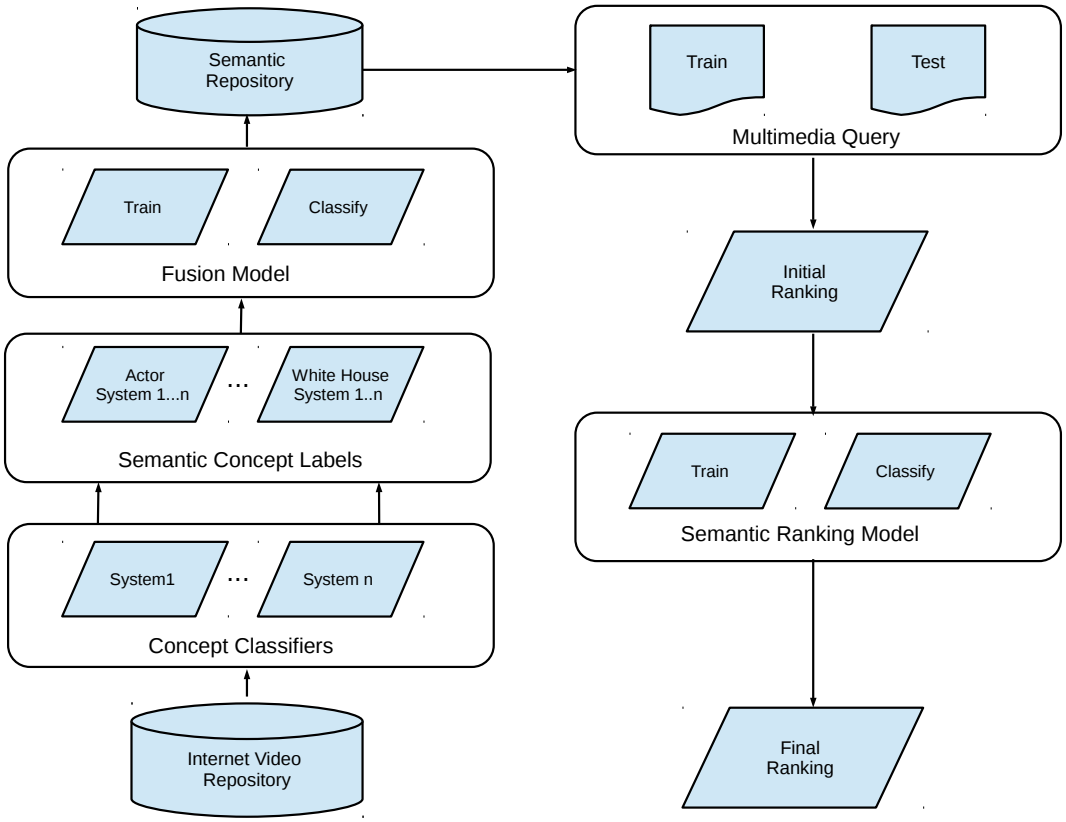


Figure 6.1: SemRank: Semantic Rank Learning

Given a set of training queries, we construct a rank learning feature space consisting of semantic features from the query, video, and their initial similarity ranking. The resulting semantic ranking model is applied to the test query set which generates a final ranking.

6.1.1 Semantic Concept Fusion

Multimedia retrieval in a semantic concept space, presents a challenge both in the quality of concept classifiers and in the number of concept labels required to provide coverage for a generic retrieval task. It is difficult to obtain labeled data for training a large number of classifiers and extracting features from a large video repository is resource intensive. Our approach fuses the output labels from different systems to provide a higher quality set of semantic concept labels.

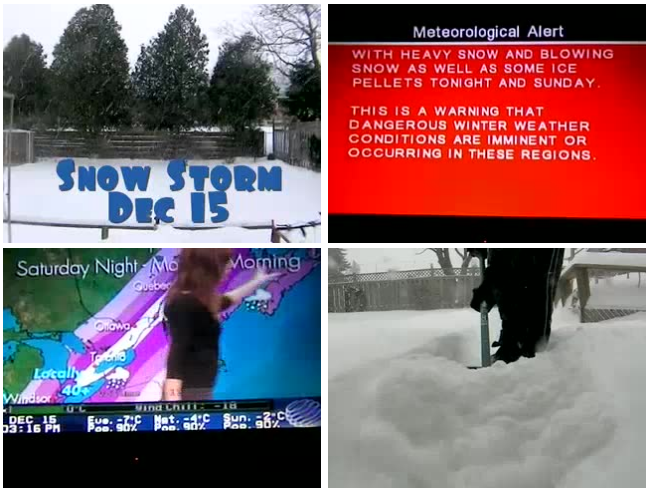
The video repository is populated with short internet videos and metadata. These multimedia objects includes modalities such as image, speech, and text, which are used to derive the input feature space for the concept classifiers. Table 6.3 provides an example video from our internet repository. The example includes speech to text, image frames, optical character recognition, and metadata provided by the video author.

We are given the semantic concept output from a number of different systems. These systems are trained with different modalities and a variety of supervised and semi-supervised learning algorithms [1]. Some of the classifiers were trained on text only while others attempted to combine features from all of the multimedia modalities. The training sets for the different systems also varied with some systems using only in-domain, while others attempted to include out-of-domain data. The output from each system is a ranked list of the videos most likely to contain the given concept. For any given video we are provided with a different set of semantic labels for each of the n systems. Table 6.4 shows an example of the different system output for a given video.

A set of classifier systems is defined as $S = (s_1, \ldots, s_n)$, where n is the number of systems providing concept labels. For a given concept c, a system s produces a ranking $R(s_c, V) = (r^c_{v_1}, \ldots, r^c_{v_m})$, where V is the video repository, s_c is the system model, and $r^c_{v_t}$ is the rank of video v_t. We define the semantic mapping of video v_t and concept c_x for all systems S, as $F_s(v_t, c_x, S) = ((r^{c_x}_{v_t})_{s_1}, \ldots, (r^{c_x}_{v_t})_{s_n})$.

A Support Vector Machine [81] is used to construct a fusion model for each semantic concept in C. Our approach fuses the output from the different systems into a single

Table 6.3: Video with metadata

Video Shots	
FileName	MMMMMoon-WinterStormDec152007277-3.
ASR	And get them in and move around the best through it seems you know you're in the home and bring against them . Yeah it is out of work...
OCR	uogi ralo Alert HEAVY suaw AND amwmzs snow wan arm ICE...
Title	Winter Storm Dec. 15, 2007.
Description	The day that was. Winter storm Watch.... at Mobile Station 1, via our on site reporter.

semantic set. The input to the SVM for a concept c_x, video v_t, and label l, is of the form $(((r^{c_1}_{v_t}, \ldots, r^{c_d}_{v_t})_{s_1}, \ldots, (r^{c_1}_{v_t}, \ldots, r^{c_d}_{v_t})_{s_n}), l)$. This approach considers the output of all concept classifiers C and all systems S when constructing a fusion model for the concept c_x. This allows the fusion model to learn relations between concepts and consider their correlations during model creation. As an example, when training a fusion model for the concept Outdoor, input to the learning algorithm includes the features from the concepts

Table 6.4: Semantic concept labels

Shot	**Semantic Labels**
	Outdoor, Plant, Road, Sky, Vegetation, ...
	Airplane, Birds, Boat Ship, Car Racing, Daytime Outdoor, Dogs, Military, Sky ...
	Weather Security Checkpoint, Sun ...

Ocean, Lake, and Mountain. The output from the semantic fusion model is used as input to the semantic rank learning model.

6.1.2 SemRank

Recent work in information retrieval has shown that supervised rank learning [35] can improve ranking results over traditional unsupervised models. Supervised ranking models not only consider similarity scores, but also incorporates features derived from both the query and document. Given the limited vocabulary available from our semantic concept labels, we believe that a rank learning approach could improve initial ranking by incorporating

the semantic features from our queries and videos. As an example, a ranking model may weight a Language Model similarity score higher for a video containing one set of semantic concepts, but choose a Probabilistic score for a different set. The ranking model also has the ability to use the semantic concepts of the query to identify query classes and produce different feature weights for each class.

Table 6.5: Semantic Feature Space

Type	**Feature**
Similarity	TFIDF Model Score Language Model Score Probabilistic Model Score Perc of concept match Total concept freq of match Total video freq of match
Query	Bag of concepts Concept count All match count
Video	Bag of concepts Concept count Shot count Shots with concepts Unique concept count Multi shot concepts

We define a ranking feature space for the video v_t and query q_i using the triple $(s_{q_i}^{v_t}, o^{q_i}, o^{v_t})$, where $s_{q_i}^{v_t}$ are similarity features, o^{q_i} are query features, and o^{v_t} are video features. Table

6.5 shows our derived semantic feature space. Similarity features are the traditional similarity scores from unsupervised retrieval models such as Term Frequency Inverse Document Frequency (TF-IDF), Language Model [55], and Probabilistic Model [21]. These features consider the interaction between a query-video pair. In addition to the scores, this class of features includes the percentage of match concepts, the total concept frequency (CF) of all matches, and the total video frequency (VF) of all matches. The difference between CF and VF is that a concept classifier produces output for every video shot boundary. This means that a given video may include multiple positive labels.

Video features are derived from the given video and are associated with the semantic concept labels from our semantic fusion model. This feature set includes total concept count, frame count, and the unique concept count. The video class also includes the bag-of-concepts, which is the semantic concept equivalent of the bag-of-words used in natural language processing. Query features include bag-of-concepts, concept count, and the count of videos matching all query concepts.

Our semantic ranking approach uses gradient boosted regression trees [68] [56] [58] to model this unique semantic feature space. The algorithm follows a step-wise approach to construct a series of N weak regression tree models, $M(s_q^v, o^q, o^v)$, over the feature vector tuple (s_q^v, o^q, o^v). At each step, i, a weight value, β_i, is learned for the given model. The final ranking is generated by combining the output, $\sum_{i=1}^{N} \beta_i \times M_i(s_q^v, o^q, o^v)$, from each of the weak models.

6.2 Experiments

We evaluate our approach using a large internet collection from the KIS and Semantic Indexing tasks of the 2012 TRECVid evaluation [1]. The IACC.1 multimedia test collection consists of approximately 8000 internet videos, available from the Internet Archive under a Creative Commons licenses. The collection includes a diverse set of content from both professional and home videos, with a duration between 10 seconds and 3.5 minutes. Many

of the videos include metadata content in the form of title, keywords, and a description. Speech-to-text is also made available as part of the evaluation.

Queries are derived from the TRECVid KIS topic set, which is based on query-by-keyword [3], where the query is presented as a text description. Our experiments follow a query-by-concept model and use a set of semantic concepts to describe the known item query. The semantic labels for each query is derived from the collaborative truth annotation task described in [44] [45]. We drop any query which does not have a concept label and evaluate our system with the remaining 328 query-by-concept topics.

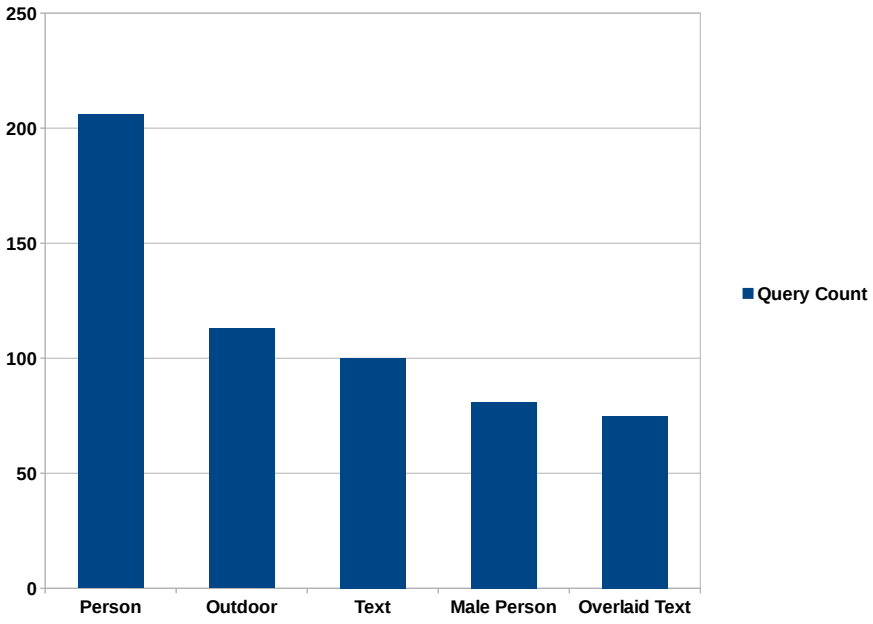


Figure 6.2: Top Query Semantic Concepts

Figure 6.2 shows the top 5 semantic concepts found in our query-by-concept set. The query test set includes 306 unique semantic concepts, where the majority of the queries use 10 or fewer concepts. Our analysis of the set identified two categories of concepts. The first category consists of approximately 100 concepts, which occurred in 10 or more queries. This general set includes concepts such Person, OutDoor, and Overlaid Text. These concept provide a general filter to identify candidate videos. The second category consists of the approximately 100 concepts that were used in 3 or fewer queries, with about 50 occurring in only 1 query. This low frequency concept category allows the system to identify the unique properties of a known item video. Examples from the low frequency category include Tent, Stadium, Skier, John Kerry, and First Lady.

Our experiments use the semantic label output from approximately 50 different systems [1]. These systems vary in the domain of data used for training and the types of training algorithms. Many of the systems were trained with only IACC training data, while others include features from external sources such as internet search results. Each system produces labels for up to 346 semantic concepts and returns a ranked list of up to 2000 video shots most likely to match a given concept.

To measure the performance of the system, we calculate the inverted rank of the truth video for a given query. The mean inverted rank is used to measure the retrieval performance over the entire query set.

Our baseline experiment follows an unsupervised information retrieval approach. We construct the semantic concept set for a given video using the output from the 50 classifier systems. A classifier labels video at the shot level, which can result in multiple labels for a given video. We capture the ranking from each system and create an aggregate across systems using the count, best, worst, and average ranks. The semantic feature vector for the baseline approach is constructed by binning the average rank for each concept. The bins allow us to create a weighted bag-of-concept words for indexing and retrieval. The query feature space uses a similar bag-of-concept words approach. A probabilistic information retrieval model is used to compare and rank the query-video pairs.

Table 6.6: Baseline Results

Rank	**MIR**	**Count**
@001	0.0091	003
@003	0.0137	007
@005	0.0160	010
@010	0.0192	018
@020	0.0217	030
@050	0.0239	054
@100	0.0250	078

The baseline results in table 6.6, show that this unsupervised approach to query-by-concept is able to identify 78 of the known items within the top 100 ranked videos. The results also show that 18 known items are ranked in the top 10 returned videos. These results are promising for our overall goal of showing that at query-by-concept approach can be successful for known item search. We were able to show that a simple retrieval model, using noisy concept labels, was able to retrieve approximately 80 of the known items.

Our next experiment builds on the baseline results using our semantic rank learning approach. We construct a semantic feature space using query-video similarity scores, query concepts, and video concepts. The similarity scores are derived from the ranked results of our unsupervised information retrieval models.

We obtain the query-video similarity scores using the output ranks from tf-idf, language [55], and probabilistic [21] models. For each of these models. we derive the percentage of concept matches, the total concept frequency of the matches, and the total video frequency of the matches. The query features are constructed using the bag-of-concepts, concept count, and the number of videos that matched all concepts in the query.

The video feature space includes: bag-of-concepts, concept count, number of shots in the video, count of shots with at least one concept, number of concepts occurring in single shot, and the number of concepts occurring in more than one shot. The weight for the bag-of-concepts in video feature space is derived from the average rank, as described in our baseline experiment. The gradient boosted regression tree model is trained using a 10-fold cross validation of the query set. The model uses 100 trees and is trained with a learning rate of .05. The maximum number of leaves for each tree is set to 5.

Table 6.7: Baseline Semantic Ranking Results

Rank	**MIR**	**Count**
@001	0.0152	005
@003	0.0183	007
@005	0.0201	010
@010	0.0246	020
@020	0.0276	035
@050	0.0309	069
@100	0.0320	096

The results in 6.7, show an improvement in count found and mean average rank over the unsupervised retrieval model. The total number of known items found, increased to 96, with 20 found in the top 10 returned results. These results show that the semantic ranking model is able to learn from our rich semantic feature space.

The baseline fusion experiment uses a supervised learning approach to derive labels for

each video concept. A semantic fusion model is constructed for each semantic concept, using a Support Vector Machine [81]. The feature space is derived from the labels of all systems, for the given video. We train the models using a 10-fold cross validation of the video collection. The labels from the semantic models are used to construct the video bag-of-concepts, which are then used in a probabilistic retrieval model.

Table 6.8: Semantic Fusion Results

Rank	**MIR**	**Count**
@001	0.0274	009
@003	0.0335	013
@005	0.0349	015
@010	0.0414	031
@020	0.0446	047
@050	0.0477	077
@100	0.0488	104

Our semantic fusion model, shown in table 6.8, improves over the baseline information retrieval model with increases in total known items found and mean inverted rank. The results show that the semantic fusion model is able to fuse the outputs from our noisy classifiers and improve retrieval results.

The final experiment applies our semantic rank learning model to the semantic fusion results. The semantic feature space is constructed in a similar manner to the baseline ranking model. The primary difference is that the similarity features are derived from the

fusion results. We maintain the gradient boost regression tree parameters and perform a 10-fold cross validation.

Table 6.9: SemRank Results

Rank	**MIR**	**Count**
@001	0.0366	012
@003	0.0478	021
@005	0.0540	030
@010	0.0595	043
@020	0.0639	064
@050	0.0663	088
@100	0.0675	118

The final results, shown in table 6.9, provide our best scores for known items retrieved and mean inverted rank. These results show that a query-by-concept model is an effective approach for multimedia KIS. The results also show that our semantic fusion model was able to improve the performance of the noisy concept labels. The semantic rank learning model improved the initial ranking results using the feature space derived from the query, video, and query-video similarity.

Chapter 7: Social Media Known Item Search

In this chapter we show that our multiview ranking approach can be extend to the problem of social media KIS. Multimedia is becoming the dominant form of communication in social media. A popular social media site [82] currently boosts over 300 million users, sharing more than 70 million images and videos every day.

Social media content includes photos and videos from daily life activities, inspirational messages, advertisements, art, leisure, travel, sports, and entertainment. Search and ranking is an import social media task, given the diverse set of multimedia content and the size of these social media collections. Users want their content to be easily accessible and they want the ability to quickly and accurately find content of interest. Users search a social media collection using short text queries consisting of keywords, phrases, or hashtags.

Table 7.1 shows an example image taken from the public feed of a popular social media site. The image content shows a picture of Rockefeller Center in New York City. Metadata content for this image includes donor provided information such as location, caption, and tag text. The metadata content also includes comments from other social media users. An example KIS query to find this item is the short text phrase, 'rockefeller center, new york, photoguy'.

Social media KIS presents a number of challenges not found in the internet video KIS problem. First, the content includes a unique set of metadata that includes geographic location information, text captions, hashtags, and comments. In addition, the metadata content includes informal text where users do not follow standard grammar practices and often omit words, punctuation, and capitalization. Finally, social media KIS queries consist of short phrases or keywords, similar to the queries submitted to major internet search engines.

Our contributions to the multimedia retrieval community include the following:

Table 7.1: Social Media Image

1. We introduce natural language processing (NLP) techniques to overcome the challenge of informal text found in social media.

2. We propose a multiview approach to model the diverse set of social media metadata.

3. Finally, we propose a social media rank learning model, and show improved ranking over traditional information retrieval models.

7.1 Approach

Our social media KIS approach follows our multiview ranking algorithm. We model each modality, from the social media metadata and visual content, as a unique view of the system. For each of the social media views, a unique feature space, defined in table 7.2, is derived from queries, videos, and query-video pairs. View specific ranking model are created using our gradient boosted regression tree approach. The output from the view specific models

Table 7.2: View Rank Features

Type	**Feature**
Query	Bag of NE Bag of Words Token Count Phrase Count
Query-Video	TFIDF Weight Model BM25 Probabilistic Model Language Model
Video	Bag of Tags Bag of NE Bag of Words Length

are combined to create the final multiview ranking. The social media ranking models follow our previous ranking approach of identifying pseudo positive KIS examples using the K-Step Markov approach.

Visual views for OCR and semantic concepts are extracted from the visual content. An analysis of the OCR extraction [83] shows that approximately 10 percent of the social media collection includes text in images or video. We found that many social media postings include inspiration text messages that can be extracted from the visual content. To identify concepts within the visual content, we use a semantic concept detector trained [84] over the ImageNet [85] collection. The deep convolutional neural network (DCNN) [86] was trained on approximately 1.2 million internet images comprising 1000 different categories. We use this DCNN model to identify semantic concepts in each image and video frame of our social media collection.

Text views for named entities, hashtags, geographic locations, text captions, and comments are derived from the metadata content. The text found within these views does not follow standard grammar rules and presents a major challenge for the social media KIS.

7.1.1 Named Entity Recognition

Named Entity Recognition (NER) [70] is an NLP technique used to locate and categorize names in text. NER was first studied in the content of formal text [87] and focused on categories of names such as person, location, and organization. Named entities are a key component of search that influence query analysis, document indexing, and retrieval models. Analysis of web search query logs [88] found that 71% of queries contain named entities. Named entities are also an important component of rank learning where their use as a query feature [63] can be used to identify classes of queries. Our ranking approach uses named entities as a feature and as a unique view in the multiview model.

Statistical NLP approaches have been traditionally applied to formal English language domains using a small set of named entity categories. NER models are trained using annotated sets of formal documents, such as the North American News Text Corpora [70]. These approaches have been able to achieve good results using features derived from text which follows former grammar rules.

Table 7.3: Informal Text

Metadata
we need to go to kaui
see u in august
Now I need to read up on Kaltenegger
"I am your father!"#StarWars=awesome

Recent NER research on informal text reported [89] a 45% decrease in F-Score when applying formally trained entity models to social media. The formal NER approach begins

to break down in this informal domain due to a shift in the writing style found in social media. Table 7.3 provides an example of informal text taken from the metadata content of our social media collection. We see in these examples, a dramatic shift in the writing style that has resulted from social media imposed character limits and smart phone text entry. Grammar, punctuation, and capitalization are often ignored in this type of informal text. In addition, we see shorter text and the creation of a new vocabulary that results from word concatenation and shortcut abbreviations.

Named entities provide an import view for our rank learning model. They are frequently found in the queries and metadata fields of our social media collection. The feature space used by the ranking model includes name categories as a query feature. Named entities also create a unique view in our model, where they provide a partial ordering of query-video pairs. In our social media collection we find named entities in the caption, location, and comments metadata. The named entity category set includes: Person, Location, Organization, URL, Email, Phone, Date, Time, Money, and Percent.

Given the metadata, $x = (x_1 \dots x_n)$, where n is the length of the text and x_i is a word in the text. We define NER over social media text as the mapping:

$$f(x_1 \dots x_n) = (y_1 \dots y_n) \tag{7.1}$$

where $y \in (y_1 \dots y_l)$ is the set of l named entity categories.

Our approach, called SM-NER, maps a given text message into a feature space that is used by the NER model. The feature set, defined in Table 7.4, consists of context, word, and gazetteer features. The word features represent normalized tokens in the training set, which are mapped as binary features to a column in the feature space. Content features identify the words to the left and right of the current word and are used to identify common patterns for a named entity. Character n-grams identify length n sub-sequences of the current word and are used to extract the current word root, suffix, and prefix. Gazetteers provide a list of name and help to improve precision for known entities.

Our NER approach uses an SVM-HMM [81] sequence tagger as the supervised classifier.

Table 7.4: NER Features

Feature	**Description**
Word	Current word
Left n-context	n words to left
Right n-content	n words to right
Char n-gram	Character n-gram
Length	Text length
Position	Current word position
Tag List	HashTag gazetteer
Location List	Location gazetteer

Given a metadata field, our approach first segments the text into a sequence of words. Each word is then mapped into a feature vector using the features defined in Table 7.4. The sequence tagger takes as input the feature vector, where each row identifies a word and the columns represent the features for that word. Labels for the training data are derived from the l named entity categories. We use a Begin-In-Out (BIO) label encoding [87] where the B and I subcategories are used to mark a multi-word named entity.

7.1.2 Word Segmentation

Hashtags have become the primary approach for social media donors to categorize content. These metadata tags are treated as keywords which can be clustered and searched within large social media collections. The hashtag format often include multiple keywords and abbreviations that are concatenated together to form a single token. Table 7.5 provides an example listing of hashtags that are found in our social media collection. The format of these hashtags presents a challenge for search and ranking in this metadata view.

To extract keywords from the hashtag view we propose a word segmentation approach based on character n-grams. Character n-grams are often used in NLP tasks [70] as a set

Table 7.5: Metadata Tags

HashTag	**Count**
costarica	303
countrymusicsinger	1
downtownnashville	34
ferretipizzaria	20
ferriswheel	41
futbolmexicano	3
washingtondczoo	1
niagrafalls	11
aclsurgery	2
carsandcoffeepalmbeach	5

of features that provide a language independent approach to extract a words root, prefix, or suffix. Given the word $w = (c_1 \dots c_t)$, where t is word length, and c_j is a character in the sequence. we define a character n-gram of length n as:

$$w' = (c_{1+i} \dots c_{n+i}), \forall 0 \leq i \leq (t-n) \tag{7.2}$$

Our word segmentation approach takes a hashtag as input and outputs all character sequences of length n.

7.2 Experiments

Our collection consists of approximately 250,000 social media posts collected from the public stream of a major social media site. The set is divided into 200,000 posts for the ranking experiments and 50,000 posts for the NER evaluation. The posts were collected over a period of two weeks during December of 2014. The ratio of images to video shared by donors is approximately 10 to 1 within the collection. The content ranges from "selfie"

images to professional advertisements. Most of the media includes metadata provided by the donor or from other users of the site.

Table 7.6: Social Media KIS Queries

KIS Query Text
blue airplane called grumm
girl white dress, fancy necklaces, jessica fang
driverlicense, social security, philippines
girl with warriors shirt
fancy wedding garter
treat me like game
girl, necklace, sitting in car
water, binoculars
water, boats, bridge, sydney cityscape
summer doll contacts
show or conference, promoting video game, chinese writing
girls in hats with letter M, spell mama with fingers
woman, photo shoot
cafe, table and chairs
desert, happy birthday on label, chinese writing

The evaluation set includes over 300 social media KIS truth queries. The queries consist of one to five short phrases which describe what the searcher remembers about the known item. Table 7.6 provides a sample of the social media KIS queries. The queries include phrases which describe one or more of the visual or metadata contents. Table 7.7

Table 7.7: Metadata Fields

Field	**Count**
Type	269,512
User	269,512
Caption	265,419
Filter	269,512
Location	33,042
HashTags	266,690
Comments	127,769

provides a listing of the metadata fields and their counts from the collection. Many of the queries include usernames, hashtags, geographical locations, and text found in the metadata comments.

Table 7.8: Final Model Results

Field	**Description**	**MIR**
Early Fusion	IR model with merged metadata	0.13
Early Fusion NLP	NLP model with merged metadata	0.24
Late Fusion	NLP Modality classification	0.25
$Multi^2Rank$	Multiview NLP Rank Learning	0.30*

We conduct a series of experiments over the social media KIS data to evaluate our multiview ranking approach. All experiments are evaluated over 100 runs using a 10-fold

cross validation of the 300 queries from the social media collection. Results are reported in table 7.8 and are calculated using the mean inverted rank. A pairwise t-test over the $Multi^2Rank$ and Late Fusion results from the social media collection, show a statistically significant difference at the 95% confidence interval.

An early fusion model [38] is used as the baseline approach. This approach does not perform hashtag segmentation or named entity recognition and results in our lowest ranking score. Our second approach, combines the early fusion model with our NLP techniques and results in a higher MIR score. An analysis of the results for this approach show that both word segmentation and social media NER helped to improve the rankings. Results for the late fusion model [4], show that this approach performs only slightly better than early fusion with NLP. We believe that the late fusion model is less effective do to the short phrases used in social media queries. Our final experiment uses our multiview NLP ranking approach and results in the best KIS rankings. The results from this series of experiments show that our NLP techniques and multiview ranking approach are effective for social media KIS.

Table 7.9: 3-Category NER F-Score

Model	**Person**	**Location**	**Organization**
Formal Text	0.49	0.56	0.46
Twitter	0.62	0.66	0.51
SM-NER	0.62	0.69	0.55

To evaluate our social media named entity approach we compare it with a formal text trained [90] model and a Twitter trained [71] model. The truth set consists of the 10 named

Table 7.10: 10-Category NER

Category	**Precision**	**Recall**	**F-Score**
Date	0.79	0.71	0.75
Email	0.80	0.78	0.79
Location	0.74	0.64	0.69
Money	0.75	0.71	0.73
Organization	0.67	0.46	0.55
Percent	0.75	0.73	0.74
Person	0.70	0.55	0.62
Telephone	0.55	0.75	0.63
Time	0.77	0.57	0.66
URL	0.90	0.82	0.86

entity categories annotated over the metadata content and comment text of 1000 posts. The formal text model is trained using the annotated collection of news wire articles from the CoNLL-2003 shared task [90]. The Twitter model is constructed from 10 thousand English tweets annotated using Amazon Mechanical Turk. Our SM-NER model is trained using the 10 thousand English tweets [71] and 5 thousand additional social media posts. Table 7.9 presents F-Score results for the models using the traditional person, location, and organization categories. We see from the results that the news wire trained CoNLL-2003 [90] model has difficulty identifying names in the informal text. The results show that the combination of Twitter and social media training data provides the best F-Score results for the SM-NER approach. The full 10 category NER results are displayed in table 7.10.

Chapter 8: Conclusions and Future Work

This thesis introduces a multiview ranking approach to the multimedia known item search task. KIS is a retrieval problem where a searcher has previously seen an image or video and would like to find it again in a repository. The searcher creates a text only query describing what they remember about the metadata and visual content of the known item. We introduce the concept of view specific phrases to mitigate the problem of view drift caused by the verbose KIS queries. A semi-supervised graph-based algorithm is used to identify pseudo-positive examples and help overcome the class imbalance problem. Finally, we model the retrieval task as a multiview rank learning problem, where each modality is treated as a unique view of the system. We evaluate our algorithms using both a benchmark dataset from the TRECVid [1] evaluation and a large social media collection of metadata and visual content.

We conclude our work with three questions, which will help guide our future research in multimedia retrieval.

1. How does the big multimedia problem impact feature space representations and models? Social media has forced us to consider algorithms that scale to hundreds of millions of objects and thousands of queries.

2. How can we automatically adapt ranking models to current trends and domains? The content of social media collections is constantly changing based on current events, politics, and region.

3. How can we model complex sequences of events within a ranking model? Increasingly, multimedia retrieval system are asked to search for sequences of activities that occur in specific places and times.

Bibliography

Bibliography

[1] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*. New York, NY, USA: ACM, 2006, pp. 321–330.

[2] Y. Press, "Youtube statistics," Accessed: 2013-08-01.

[3] C. G. Snoek and M. Worring, "Concept-based video retrieval," *Foundations and Trends in Information Retrieval*, vol. 2, no. 4, pp. 215–322, 2008.

[4] K.-W. Wan, Y.-T. Zheng, and L. Chaisorn, "Known-item video search via query-to-modality mapping," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 1133–1136.

[5] J. Cao, Y.-D. Zhang, L. Pang, B. Feng, and J.-T. Li, "Known-item search by mcg-ict-cas." in *TRECVID*, 2010.

[6] H. Li *et al.*, "Informedia at trecvid2010." in *TRECVID*, 2010.

[7] M. Mitra, A. Singhal, and C. Buckley, "Improving automatic query expansion," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998, pp. 206–214.

[8] P. Salembier, "Overview of the mpeg-7 standard and of future challenges for visual information analysis," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 1, pp. 343–353, 2002.

[9] D. K. Park, Y. S. Jeon, and C. S. Won, "Efficient use of local edge histogram descriptor," in *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*. New York, NY, USA: ACM, 2000, pp. 51–54.

[10] Y.-G. Jiang, C.-W. Ngo, and J. 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*. New York, NY, USA: ACM, 2007, pp. 494–501.

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2003, p. 1470.

[12] M. Campbell *et al.*, "Ibm research trecvid-2007 video retrieval system," in *In the proceedings of TREC Video*. NIST, 2007.

[13] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis, "Large-scale concept ontology for multimedia," *Multimedia, IEEE*, vol. 13, no. 3, pp. 86–91, 2006.

[14] M. Huijbregts, R. Ordelman, and F. de Jong, "Annotation of heterogeneous multimedia content using automatic speech recognition," in *Proceedings of the Second International Conference on Semantic and Digital Media Technologies, SAMT 2007*, ser. Lecture Notes in Computer Science, vol. 4816, Berlin, 2007, pp. 78–90.

[15] C. G. M. Snoek and M. Worring, "Multimodal video indexing: A review of the state-of-the-art," *Multimedia Tools Appl.*, vol. 25, no. 1, pp. 5–35, 2005.

[16] M. esther Vidal, "Wordnet: An electronic lexical database," in *In Proceedings of the ACM SIGMOD Workshop on The Web and Databases (WebDB)*. MIT Press, 1998.

[17] "Wikipedia, the free encyclopedia," http://www.wikipedia.org.

[18] L. Rigutini, E. Di Iorio, M. Ernandes, and M. Maggini, "Automatic term categorization by extracting knowledge from the web." in *ECAI*, G. Brewka, S. Coradeschi, A. Perini, and P. Traverso, Eds. IOS Press, 2006, pp. 531–535.

[19] P. Wang and C. Domeniconi, "Building semantic kernels for text classification using wikipedia," in *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2008, pp. 713–721.

[20] J. Liu, W. Lai, X.-S. Hua, Y. Huang, and S. Li, "Video search re-ranking via multi-graph propagation," in *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*. New York, NY, USA: ACM, 2007, pp. 208–217.

[21] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *Journal of the American Society for Information science*, vol. 27, no. 3, pp. 129–146, 1976.

[22] J. Tešić, A. P. Natsev, and J. R. Smith, "Cluster-based data modeling for semantic video search," in *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*. New York, NY, USA: ACM, 2007, pp. 595–602.

[23] T. Westerveld and A. P. D. Vries, "Multimedia retrieval using multiple examples," in *International Conference on Image and Video Retrieval*, 2004.

[24] A. P. Natsev, M. R. Naphade, and J. R. Smith, "Semantic representation: search and mining of multimedia content," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2004, pp. 641–646.

[25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[26] M. R. Naphade *et al.*, "A light scale concept ontology for multimedia understanding for trecvid 2005," 2005.

[27] L. Xie, A. Natsev, and J. Tesic, "Dynamic multimodal fusion in video search," in *ICME*, 2007, pp. 1499–1502.

[28] R. B. N. Aly, D. Hiemstra, and R. J. F. Ordelman, "Building detectors to support searches on combined semantic concepts," in *Proceedings of the Multimedia Information Retrieval Workshop, Amsterdam, The Netherlands.* Amsterdam: Yahoo! Research, August 2007, pp. 40–45.

[29] S. Wei *et al.*, "Bjtu trecvid 2007 video search," in *TRECVID 2007.* NIST, 2007.

[30] P. Wilkins, P. Ferguson, and A. F. Smeaton, "Using score distributions for query-time fusion in multimediaretrieval," in *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval.* New York, NY, USA: ACM, 2006, pp. 51–60.

[31] M. Richardson and P. Domingos, "The intelligent surfer: probabilistic combination of link and content information in pagerank," in *In Advances in Neural Information Processing Systems.* MIT Press, 2002, pp. 1441–1448.

[32] H. M. Blanken, A. P. de Vries, H. E. Blok, and L. Feng, *Multimedia Retrieval (Data-Centric Systems and Applications).* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[33] E. Yilmaz and J. A. Aslam, "Estimating average precision with incomplete and imperfect judgments," in *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management.* New York, NY, USA: ACM, 2006, pp. 102–111.

[34] Y.-G. Jiang, A. Yanagawa, S.-F. Chang, and C.-W. Ngo, "CU-VIREO374: Fusing Columbia374 and VIREO374 for Large Scale Semantic Concept Detection," Columbia University, Tech. Rep., August 2008.

[35] T.-Y. Liu, "Learning to rank for information retrieval," *Foundations and Trends in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[36] C. Burges, "From ranknet to lambdarank to lambdamart: An overview," *Microsoft Research Technical Report*, vol. MSR-TR-2010-82, 2010.

[37] N. Chen, J. Zhu, and E. P. Xing, "Predictive subspace learning for multi-view data: a large margin approach," in *Advances in neural information processing systems*, 2010, pp. 361–369.

[38] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[39] C.-J. Lee, W. B. Croft, and J. Y. Kim, "Evaluating search in personal social media collections," in *Proceedings of the fifth ACM international conference on Web search and data mining.* ACM, 2012, pp. 683–692.

[40] J. Kim and W. B. Croft, "Ranking using multiple document types in desktop search," in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2010, pp. 50–57.

[41] N. Craswell, A. P. de Vries, and I. Soboroff, "Overview of the trec-2005 enterprise track," in *TREC 2005 conference notebook*, 2005, pp. 199–205.

[42] L. Chaisorn, Y.-T. Zheng, and K. Sim, "Known-item search (kis) in video: Survey, experience and trend," in *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*. IEEE, 2011, pp. 1–4.

[43] S. Chen, K. McGuinness, R. Aly, N. E. O'Connor, and F. de Jong, "The axes-lite video search engine," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2012 13th International Workshop on*. IEEE, 2012, pp. 1–4.

[44] S. Ayache and G. Quénot, "Video corpus annotation using active learning," in *Advances in Information Retrieval*. Springer, 2008, pp. 187–198.

[45] M. Hradiš, M. Kolář, A. Láník, J. Král, P. Zemčík, and P. Smrž, "Annotating images with suggestionsuser study of a tagging system," in *Advanced Concepts for Intelligent Vision Systems*. Springer, 2012, pp. 155–166.

[46] J. Guo, Z. Qiu, and C. Gurrin, "Exploring the optimal visual vocabulary sizes for semantic concept detection," in *Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on*. IEEE, 2013, pp. 109–114.

[47] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[48] A. P. Natsev, A. Haubold, J. Tešić, L. Xie, and R. Yan, "Semantic concept-based query expansion and re-ranking for multimedia retrieval," in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 991–1000.

[49] J. Dalton, J. Allan, and P. Mirajkar, "Zero-shot video retrieval using content and concepts," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013, pp. 1857–1860.

[50] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[51] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen, "Multiview vector-valued manifold regularization for multilabel image classification," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 5, pp. 709–722, 2013.

[52] R. Yan and M. Naphade, "Semi-supervised cross feature learning for semantic concept detection in videos," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 657–663.

[53] J. Yu, Y. Rui, and B. Chen, "Exploiting click constraints and multi-view features for image re-ranking," *Multimedia, IEEE Transactions on*, vol. 16, no. 1, pp. 159–168, Jan 2014.

[54] Z. Cao *et al.*, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 129–136.

[55] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998, pp. 275–281.

[56] Y. Ganjisaffar, R. Caruana, and C. V. Lopes, "Bagging gradient-boosted trees for high precision, low variance ranking models," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011, pp. 85–94.

[57] Q. Wu, C. J. Burges, K. M. Svore, and J. Gao, "Ranking, boosting, and model adaptation," *Tecnical Report, MSR-TR-2008-109*, 2008.

[58] O. Chapelle and Y. Chang, "Yahoo! learning to rank challenge overview." *Journal of Machine Learning Research-Proceedings Track*, vol. 14, pp. 1–24, 2011.

[59] P. Li, Q. Wu, and C. Burges, "Mcrank: Learning to rank using multiple classification and gradient boosting," in *Advances in neural information processing systems*, 2007, pp. 897–904.

[60] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," *Advances in neural information processing systems*, pp. 115–132, 1999.

[61] C. Burges *et al.*, "Learning to rank using gradient descent," in *Proceedings of the 22nd International Conference on Machine learning*. ACM, 2005, pp. 89–96.

[62] C. Quoc and V. Le, "Learning to rank with nonsmooth cost functions," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 19, pp. 193–200, 2007.

[63] C. Macdonald, R. L. Santos, and I. Ounis, "On the usefulness of query features for learning to rank," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 2559–2562.

[64] T. Qin, T.-Y. Liu, J. Xu, and H. Li, "Letor: A benchmark collection for research on learning to rank for information retrieval," *Information Retrieval*, vol. 13, no. 4, pp. 346–374, 2010.

[65] T. Mei, Y. Rui, S. Li, and Q. Tian, "Multimedia search reranking: A literature survey," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 38, 2014.

[66] B. Safadi and G. Quénot, "Re-ranking by local re-scoring for video indexing and retrieval," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 2081–2084.

[67] L. S. Kennedy and S.-F. Chang, "A reranking approach for context-based concept fusion in video indexing and retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 333–340.

[68] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[69] S. White and P. Smyth, "Algorithms for estimating relative importance in networks," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 266–275.

[70] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[71] D. Etter, F. Ferraro, R. Cotterell, O. Buzek, and B. Van Durme, "Nerit: Named entity recognition for informal text," *Human Language Technology Center of Excellence, Johns Hopkins*, vol. Technical Report 11, 2013.

[72] J. Gauvain, L. Lamel, and G. Adda., "The limsi broadcast news transcription system." in *Speech Communication*, 2002, pp. 37(1–2):89–108.

[73] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[74] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma, "Terrier: A High Performance and Scalable Information Retrieval Platform," in *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.

[75] C.-J. Lee, Y.-C. Lin, R.-C. Chen, and P.-J. Cheng, "Selecting effective terms for query formulation," in *Information Retrieval Technology*. Springer, 2009, pp. 168–180.

[76] C. Lioma and I. Ounis, "Examining the content load of part of speech blocks for information retrieval," in *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, 2006, pp. 531–538.

[77] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 173–180.

[78] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," in *In Proceedings of the ACL conference*. Citeseer, 2013.

[79] Y.-G. Jiang, J. Wang, S.-F. Chang, and C.-W. Ngo, "Domain adaptive semantic diffusion for large scale context-based video annotation," in *International Conference on Computer Vision (ICCV)*, Kyoto, Janpan, September 2009.

[80] M.-C. De Marneffe, B. MacCartney, C. D. Manning *et al.*, "Generating typed dependency parses from phrase structure parses," in *Proceedings of LREC*, vol. 6, no. 2006, 2006, pp. 449–454.

[81] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural svms," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.

[82] "Instagram news blog," http://blog.instagram.com/post/104847837897/141210-300million.

[83] R. Smith, "An overview of the tesseract ocr engine." in *ICDAR*, vol. 7, no. 1, 2007, pp. 629–633.

[84] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[85] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," 2014.

[86] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[87] R. Grishman and B. Sundheim, "Message understanding conference-6: A brief history." in *COLING*, vol. 96, 1996, pp. 466–471.

[88] J. Guo, G. Xu, X. Cheng, and H. Li, "Named entity recognition in query," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 267–274.

[89] X. Liu, S. Zhang, F. Wei, and M. Zhou, "Recognizing named entities in tweets," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 359–367.

[90] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 142–147.

Curriculum Vitae

David Etter is a researcher in the field of Multimedia Retrieval and Natural Language Processing. He received a Bachelor of Arts in Mathematics from Shippensburg University in 1994. He obtained a Master of Science in Computer and Information Sciences from Hood College in 2001. David is currently a PhD student at George Mason University in the Department of Computer Science.