10-31-2014

Text Analytics of Social Media: Sentiment Analysis, Event Detection and Summarization

Chao Shen
cshen001@cs.fiu.edu

DOI: 10.25148/etd.FI14110776

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

TEXT ANALYTICS OF SOCIAL MEDIA: SENTIMENT ANALYSIS, EVENT DETECTION AND SUMMARIZATION

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Chao Shen

2014

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Chao Shen, and entitled Text Analytics of Social Media: Sentiment Analysis, Event Detection and Summarization, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Shu-Ching Chen

Debra VanderMeer

Jinpeng Wei

Bogdan Carbunar

Tao Li, Major Professor

Date of Defense: October 31, 2014

The dissertation of Chao Shen is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Dean Lakshmi N. Reddi
University Graduate School

Florida International University, 2014

DEDICATION

To my family.

ACKNOWLEDGMENTS

There are so many to thank. First and foremost I want to thank my advisor, Professor Tao Li. Without his encouragement and guidance, I would not have spent five enjoyable years at FIU, and this dissertation would not have existed. He is one of the rare advisors that students dream that they will find. I am grateful that he always stays with me in the best and worst moments of my Ph.D journey. In the same vein, I want to thank Professor Shu-ching Chen, Professor Debra VanderMeer, Professor Jinpeng Wei and Professor Bogdan Carbunar for being my doctoral committee. They have provided me many valuable questions and useful suggestions for my dissertation. I extend my warmest thanks to Dr. Fei Liu and Mr. Fuliang Weng in Bosch Research and Development Center, and Dr. Jian Yin in Pacific Northwest National Laboratory, who gave me help and support during my summer internships. And I would also like to thank all other my coauthors and labmates. It was my great honor to work with them. Special thanks to all my friends in Miami and the Bay Area for giving me joy and good memories in these years. Deepest graduate to my family. I am indebted to my parents, my father, Datian Shen and especially to my mother, Limin Ding. She recently passed away after fourteen-year brave fight against cancer. I would like to thank my wife, Lin Ye, for her love, support, and understanding. I love you.

ABSTRACT OF THE DISSERTATION

TEXT ANALYTICS OF SOCIAL MEDIA: SENTIMENT ANALYSIS, EVENT DETECTION AND SUMMARIZATION

by

Chao Shen

Florida International University, 2014

Miami, Florida

Professor Tao Li, Major Professor

In the last decade, large numbers of social media services have emerged and been widely used in people's daily life as important information sharing and acquisition tools. With a substantial amount of user-contributed text data on social media, it becomes a necessity to develop methods and tools for text analysis for this emerging data, in order to better utilize it to deliver meaningful information to users.

Previous work on text analytics in last several decades is mainly focused on traditional types of text like emails, news and academic literatures, and several critical issues to text data on social media have not been well explored: 1) how to detect sentiment from text on social media; 2) how to make use of social media's real-time nature; 3) how to address information overload for flexible information needs.

In this dissertation, we focus on these three problems. First, to detect sentiment of text on social media, we propose a non-negative matrix tri-factorization (tri-NMF) based dual active supervision method to minimize human labeling efforts for the new type of data. Second, to make use of social media's real-time nature, we propose approaches to detect events from text streams on social media. Third, to address information overload for flexible information needs, we propose two summarization framework, dominating set based summarization framework and learning-to-rank based summarization framework. The dominating set based summarization framework can be applied for different types

of summarization problems, while the learning-to-rank based summarization framework helps utilize the existing training data to guild the new summarization tasks. In addition, we integrate these techneques in an application study of event summarization for sports games as an example of how to better utilize social media data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

With the popularity of Internet, the volume of online text documents (e.g., news and web pages) are explosively growing. Text analytics such as document classification, clustering and summarization are developed to discover useful and meaningful information from textual documents, for users to better understand the textual datasets. For example, document clustering provides an efficient way in organizing web search results, and document summarization can generate informative snippets to help users in web exploring.

In the last decade, large numbers of social media services have emerged and been widely used in people's daily life as important information sharing and acquisition tools. New characteristics of text on social media impose challenges to the traditional text analytics, which is focused on conventional text like news and general web pages. My research goal is to develop text analytics for text data on social media by addressing its differences from the conventional text data, to help users to better understand and utilize a large volume of social media text. In particular, we focus on three dimensions: sentiment analysis, event detection and summarization, more specifically by answering the following questions:

Sentiment analysis *How to quickly train a sentiment analysis model for text on social media with minimum human effort?* Sentiment analysis is a critical step to understand people's preference and feelings from social media data. Most of the sentiment analysis methods assume availablity of training data. Since existing models and tools trained on traditional text are not applicable to text on social media due to the big differences in

language usage, new training data has to be labeled, which is a costly process. So we need to find an effective way to label data to reduce human effort to minimum.

Event detection *How to detect events discussed in social media and associated posts?* Because of social media's real-time nature, large number of event-related posts exist on social media, and can be used to update social media users and the public on what events are happening in the world. Event detection aims to identify these events and their associated posts, so that information about an event discussed on social media can be well organized and presented to users.

Multi-document summarization *How to generate a summary aggregating information from a large set of textual posts on social media for flexible information needs?* Multi-document summarization is typical a tool to overcome information overload. However because of heterogeneous topics and purposes of posts on social media, users may impose different information needs, which requires different summaries of a set of textual posts from different aspects.

1.2 Background

Social media is typical known as online services for interaction among people by creating, sharing and exchanging information and ideas in real or virtual social networks [ABHH08]. It includes blogs and microblogs (e.g., Twitter[1]), content sharing communities (e.g., Flickr[2], YouTube[3]) social networks (e.g., Facebook[4]) and etc. In the last decade, these social

[1]http://www.twitter.com

[2]http://www.flickr.com

[3]http://www.youtube.com

[4]http://www.facebook.com

media sites are becoming increasingly popular and important information distribution tools for users to share their statuses, experiences and interests. Consequently, substantial amounts of user-contributed materials (e.g., photographs, videos, and textual content) are constantly being uploaded to these sites of a wide variety of topics.

Although current social media is enriched with multi-media content like images and videos, text is still one of the most important types of content, which can be used alone as in most posts on Twitter and Facebook, or as descriptions and comments of photographs and videos. In order to provide better services and deliver meaningful information to users of social media and the public, it is imperative to create tools to conduct fundamental text analysis to better understand and obtain basic information from a large volume of textual posts on social media.

Social Media Type	Typical Examples
collaborative projects	Wikipedia
blogs and microblogs	Twitter
social news networking sites	Digg, Leakernet
content communities	YouTube, DailyMotion
social networking sites	Facebook
virtual game-worlds	World of Warcraft
virtual social worlds	Second Life

Table 1.1: A classification scheme created by [KH09].

According to the classification scheme created by [KH09] described in Table1.1, there are seven major types of social medias. In this dissertation, we are more focused on three of them: blogs and microblog, content communities, and social networking sites, which are in italic in Table1.1. Text information on these three types of social medias plays an important role, and have the following fundamental differences compared to traditional text:

- Text on social media is rich in sentiment information. It's very common that people express likes and dislikes through posts like status updates and comments. Thus so-

cial media is a source of crowd intelligence that can be used to investigate common feelings about some particular topics.

- Text on social media carries a lot of real-time information. "What's happening?" is a typical question that users of social media answer by new posts. People report or publish comments on the events they are experiencing of a wide variety of types and scales around the world, ranging from a natural disasters to a sports game.
- Text on social media is heterogenous and large in volume. Varieties of tools like applications of mobile devices enable users to easily generate and share content on social media sites. Consequently, a large volume of text data, which serves different purposes, is created over a wide range of topics.

Because of these differences, it is not applicable to simply adapt existing text analysis techniques of traditional text data to social media data.

There have been many studies on social networks, which are the background structure behind social media, from fundamental research on the properties of a social network [Kle00, KKT03] to applications like communities detection [LNK07, GN02, LLM10], influential users identification [CHBG10, TSWY09, AW12], information diffusion [YL10, GGLNT04, YC10], and social network evolution [BJN+02, KW06]. For text analytics, while many existing techniques are developed for traditional text like emails, news and academic documents, recent studies extend them to social media text by incorporating information of the background social network [WLJH10, CWML13, CNN+10].

1.3 Contribution of This Dissertation

In this dissertation, we focus on developing effective methods for the following three aspects corresponding to the three aforementioned characteristics of text on social media (1) learning a sentiment analysis model for text on social media with minimum human

effort via active dual supervision from samples and features, (2) detecting events from a social media stream, and (3) summarizing documents for various summarization tasks for the flexible information needs from social media data. In the dissertation, a real-time application of sports game summarization and analysis system using Twitter streams is also presented integrating the developed techniques to demonstrate their usage in a real case.

Active Learning with Dual Supervision for Sentiment Analysis We propose a new active dual supervision approach, in which a classification model is learned actively using labels of both samples and features for sentiment analysis [SL11b]. We first extend the constrained non-negative tri-factorization framework, which incorporates labels of posts and words as constraints, to explicitly model the corresponding relationships between post classes and word classes. Then by making use of the reconstruction error criterion in matrix factorization, we propose a unified scheme to evaluate the value of post and word labels. Instead of comparing the estimated performance increase of new post labels or word labels, our proposed scheme assumes that a better supervision (a post label or a word label) should lead to a more accurate reconstruction of the original data matrix.

Participant Based Time-Content Mixture Model for Event Detection We propose a participant-based method to detect important moments along a social media stream [SLWL13]. Instead of detecting important moments directly, we first dynamically identify participants, which are named entities frequently mentioned in the input stream, then "zooms-in" the whole stream to the participant level. To detect important moments related to each participant, we propose a time-content mixture model considering both volume changes and topic changes along the stream, so that associated posts of an event are not only temporally bursty but also topically coherent. Important moments detected for different

participants, if they are close enough, can be combined based on their co-occurrence to get final events in the whole stream.

New Multi-document Summarization Frameworks for Flexible Information Requirements First we propose a multi-document summarization framework based on minimum dominating set for various summarization tasks [SL10]. The framework is originated for *generic summary*, and can be extended for several other types of summarization like *query-focused summarization*, *update summarization* and *comparative summarization*. For the *query-focused summarization*, we further propose a learning to rank based summarization framework to allow users to define the information need using the training data [SL11a].

Application: Event Summarization for Sports Games Using Twitter Streams In this application study we propose to build an event summarization application for sports games using Twitter streams, which provides an alternative way to be kept informed of the progress of a sports game and audience's responds from social media data. The application integrates the aforementioned text analysis techniques. Based on the event detection results, summarization and sentiment analysis are employed to summarize the game's progress and audience's supports for different levels: an event, a participant and the whole game.

1.4 Dissertation Outline

The rest of the dissertation is organized as follows: Chapter 2 reviews the related work. Chapter 3 proposes an approach for sentiment analysis with active dual supervision. Chapter 4 improves event detection on social media streams by integrating changes of data volume and content. Chapter 5 describes two summarization frameworks, the frame-

work based on minimum domination set for various document summarization, and the framework based on learning to rank for query-focused summarization with training data. Chapter 6 presents a real-time event summarization and analysis system for sports games integrating event detection, sentiment analysis and summarization techniques. Finally, Chapter 7 concludes the dissertation with future work.

CHAPTER 2

RELATED WORK

2.1 Preprocessing of Social Media Text

The original form of text is a string or a sequence of characters, which needs natural language processing (NLP) techniques to extract information and relations for upper layer text analysis like text mining and text retrieval. The most frequently used NLP techniques for English, which this dissertation is focused on, include: *Sentence Splitting*, which divides the whole text document into a list of sentences, *Tokenization*, which further divides text of a sentence into a list of words or tokens, *Part-of-speech (POS) Tagging*, which assigns to every word in a sentence a Part-of-Speech tag, *Shallow Parsing or Chunking*, which identifies unembedded noun, verb and adjective phrases in a sentence, and *Named Entity Recognition (NER)*, which recognizes named entities of predefined types like person, location and organization in a sentence.

A number of toolkits are available for these NLP tasks as preprocessing of conventional text data. The most widely used NLP toolkits include GATE [Cun02], OpenNLP [Bal05], and Stanfard NLP [TKMS03, FGM05]. GATE is a general architecture for text engineering for a wide variety of purposes of text analysis including annotation and semantic engineering, but its core module is an extendable rule based annotation system with a set of rules to conduct these preprocessing tasks. Both OpenNLP and Stanford NLP are learning based systems, and conduct these NLP tasks as sequential labeling problems. OpenNLP employs maximum entropy models as the learning model for all these tasks, while Stanfard NLP uses a maximum entropy model for POS tagging and conditional random fields models for shallow parsing and NER recognition. Besides the toolkits, POS tagging, shallow parsing and NER recognition attract in the last decades many researchers to propose methods for better performance in term of accu-

racy or speeds. The state-of-the-art methods are learning based using conditional random fields [LMP01, SMR07, SP03, ML03, JWL+06]. All these learning based methods need a large annotated dataset for the training purpose, and the Penn Treebank (PTB) [MMS93], which is composed of annotated news articles from Wall Street Journal, is the most widely used one for conventional text in English.

With the popularity of social media, social media text, especially short posts and comments in Facebook and microblogs in Twitter, imposes challenges and requires new methods. Comparing with conventional news text, social media text is short in length, written often in an informal language style, and contains a lot of noises. Some work has been done on POS tagging English tweets. [FCW+11] annotated a small treebank of 519 sentences from Twitter, using the PTB annotation scheme. They reported a POS tagging accuracy of 84,1% for an SVM-based tagger. TwitterNLP [RCME11] is a CRF-based tagger to Twitter data with a tagging accuracy of 88,3% using the full 45 tags from the PTB and 4 additional tags for twitter-specific phenomena (retweets, at-mentions, hashtags and urls). Ark-Tweet-NLP [GSO+11, OOD+13] is a fast tagger performing coarse-grained analysis for English microblogs with an accuracy around 92%. [OOD+13] also trained and tested their tagger on the annotated data of [RCME11] and reported an accuracy of around 90% on the 45 PTB tags plus the 4 (unambiguous) twitter-specific tags. Ark-Tweet-NLP mostly benefits from word clustering of unlabelled Twitter data using the latent Dirichlet allocation (LDA) [BNJ03]. [Reh13] extended Ark-Tweet-NLP to POS tagging for German.

2.2 Multi-document Summarization

As a fundamental and effective tool for document understanding and organization, multi-document summarization enables better information service by creating concise and in-

formative reports for a large collection of documents. Specifically, in multi-document summarization, given a set of documents as input, the goal is to produce a condensation (i.e., a generated summary) of the content of the entire input set [JM08]. The generated summary can be generic where it simply gives the important information contained in the input documents without any particular information needs or query/topic-focused where it is produced in response to a user query or related to a topic [JM08, Man01]. For the last over two decades, multi-document summarization has attracted attention of a large number of researchers, and various aspects of the problem have been explored and many methods proposed.

For generic summarization, a saliency score is usually assigned to each sentence and then the sentences are ranked according to the saliency score. The scores are usually computed based on a combination of statistical and linguistic features. MEAD [RJST04] is an implementation of the centroid-based method where the sentence scores are computed based on sentence-level and inter-sentence features. SumBasic [NV05] showed that the frequency of content words alone can also lead good summarization results. Graph-based methods [ER04, WYX07b] have also been proposed to rank sentences or passages based on the PageRank algorithm or its variants. For example, LexPageRank [ER04] constructed a sentence connectivity matrix and computed sentence importance based on an algorithm similar to PageRank, and [WYX07b] used an iterative reinforcement algorithm on sentence-sentence graph, word-word graph and sentence-word graph to extract summary and keywords simultaneously.

In comparison to generic document summarization, query-focused summarization requires a summarizer to incorporate user declared queries. The generated summary should not only reflect the important concepts in the documents but also bias to the queries. There are many recent studies on query-focused document summarization. Maximal Marginal Relevance(MMR) has been used in a document summarization system for redundancy

removal [GMCK00], in which the best sentence is considered the one that is most similar to the query and least similar to the text that is already in the summary. A non-negative matrix factorization (NMF) based query-focused summarization method was proposed in [WLZD08], which used the cosine similarity measure between the expanded query and the semantic features obtained by NMF to rank sentences. Manifold ranking was applied [WX09] to decide the relationship between the given query and the sentences by making use of the relationship among all the sentences in the documents. Probability models have also been proposed under different assumption on the generation process of the documents and the queries [DIM06, HV09, TYC09]. A recent work [Wan09] conducted subtopic analysis for document summarization, in which explicit or implicit subtopics are discovered using heuristic syntactic rules and term co-occurrence.

2.3 Event Detection

The concept of event detection is first introduced by Topic detection and tracking (TDT), which is a research program initiated by DARPA (Defense Advanced Research Projects Agency) for finding and following the new events in streams of broadcast news stories[1]. TDT consists of three major technical tasks, including the detection of unknown events, the tracking of known events, and segmentation of a news source into stories. Many promising research studies have arisen during the TDT evaluations, specifically within the information retrieval and natural language processing communities [YPC98, APL98, All02, KA04]. Most of them assume that all the documents in the given collections are somehow related to a number of undiscovered events, and which can be discovered by using text classification and text clustering techniques.

[1] http://projects.ldc.upenn.edu/TDT/

Attempts have been made to adapt the methods developed on formal document collections to event detection on social media. For example, [POL10] proposed an algorithm based on locality-sensitive hashing for detecting new events from a stream of Twitter posts. However, the assumption that all the documents in the given collections are related to a number of events is not held on social media, since the related social media posts about an event can easily be overwhelmed by a large volume of trivial ones. So most recent studies are trying to address this issue. [BNG11] proposed an online clustering technique to group together the topically similar tweets and used a SVM classifier to distinguish between the event and non-event clusters. [OKA10] proposed demo systems to display the event-related themes and popular tweets, allowing the users to navigate through their topic of interest. [ZZWV11] described an effort to perform data collection and event recognition despite various limits to the free access of Twitter data. [DJZL12] integrated both temporal information and users' personal interests for bursty topic detection from the microblogs. [RMEC12] described an open-domain event-extraction and categorization system, which extracts an open-domain calendar of significant events from Twitter.

Event detection has also been applied in summarization of social media streams, where important events are first detected as parts of the summary. [MBB+11] introduced a "TwitInfo" system to visually summarize and track the events on Twitter. They proposed an automatic peak detection and labeling algorithm for the social streams. [CP11] proposed an event summarization algorithm based on learning an underlying hidden state representation of the event via hidden Markov models. [NMD12, ZSAG12] focused on real-time event summarization, which detected the sub-events by identifying those moments where the tweet volume has sharp increases, then used various weighting schemes to perform tweet selection and finally generates the event summary.

2.4 Sentiment Analysis

A typical problem in sentiment analysis is classifying a piece of text into "Positive", "Negative" or "Neutral". "Positive" means that the user expresses the support or likeness of the target topic; "Negative" means the opposite; "Neutral" means that the text is objective. Traditionally, the classification is conducted on reviews (including blogs and comments). Various methods have been proposed to train a model for reviews of a particular domain of products given existing labeled reviews [Gam04, PLV02, WWH05, MC04].

Now with the popularity of social network such as Facebook and Twitter, many people express their opinions and comments about products, companies, politicians and events on these social media sites. As social media has become an important data source for companies to get feedback and for public affair persons to analysis the dynamic sentiment trends on public events, researchers have been working on how to adapt sentiment classification to social media data, especially Twitter data. The key issue is training data. With a large range of topics discussed on Twitter, it would be difficult to label enough social media posts for each of topics manually. In order to generate automatically training data, Twitter tags and smileys were utilized in [DTR10, GBH09]. Similar ideas were applied in [ZGD+11], where first a lexicon-based method was applied to generate high precision low recall labels, and then these labels were used to training a learning-based model to boost recall. Instead of using lexical "distant supervision" [GBH09], [ZGD+11] made use of existing twitter sentiment services like Twendz[2], Twitter Sentiment[3] and TweetFeel[4] for labels, trained several models, each based on one data source, and finally ensembled the classification results to reduce the bias and noise introduced by the training

[2]http://twendz.waggeneredstrom.com/

[3]http://twittersentiment.appspot.com/

[4]http://www.tweetfeel.com/

data. We can see that the supervision may come from lexicons in tweets, such as tags and smileys, as well as from biased tweet labelers. To leverage both types of supervision into a unified approach, dual supervision learning [SHM09] can be used. One of the methods is to conduct non-negative matrix tri-factorization (tri-NMF), mapping both tweets and terms in tweets into sentiment space, with as constraints a prior of labeled tweets and terms [LZS09].

Another issue is that unlike traditionally studied reviews, social media posts are not well organized with respect of target topics, which is important because in traditional sentiment analysis study, it has been shown that the different topic domains need different classification models. [JYZ+11] introduced target-dependent features for sentiment analysis in Twitter, so given different target, features of a tweet may be different. [DWT+14] integrated target information with a deep learning model, Adaptive Recursive Neural Network, which automatically propagates sentiments of words towards the target. But still in their work, training data was manually labeled, so the problem was only partially solved unless we can effectively reduce the cost to generate training data.

Although the difference between social media text and traditionally studied reviews imposes challenges for sentiment analysis, the social network structure behind the social media can be utilized. [TLT+11] studied user-level sentiment, analyzing sentiment over all tweets posted by a user about a target and assuming that close users share sentiment. [HTTL13] presented a mathematical optimization formulation that incorporated the sentiment consistency over social network into the supervised learning process.

CHAPTER 3

TRI-NMF BASED ACTIVE DUAL SUPERVISION

3.1 Introduction

With the popularity of social network, many people express their opinions on social media sites, like Facebook and Twitter. The large number of such posts makes social media rich in sentiment and become an important sentiment data source. In order to utilize such information on social media, it is a necessity to conduct sentiment analysis to classify a post into "Positive", "Negative" or "Neutral". Even although sentiment analysis has been well explored on text of product reviews [Gam04, PLV02, WWH05, MC04], it is still challenging on social media, since with a wide range of topics discussed on social media, it would be difficult to labeled enough posts for each of topics manually.

The challenge can be partially addressed by active learning, as an effective paradigm to optimize the learning benefit from domain experts' feedback and to reduce the cost of acquiring labeled examples for supervised learning, has been intensively studied in recent years [MN98, TK02, Set09]. Traditional approaches for active learning query the human experts to obtain the labels for intelligently chosen data samples. However, in text classification where the input data is generally represented as document-word matrices, human supervision can be obtained on both documents and words. For example, in sentiment analysis of product reviews, human labelers can label reviews as positive or negative, they can also label the words that elicit positive sentiment (such as "sensational" and "electrifying") as positive and words that evoke negative sentiment (such as "depressed" and "unfulfilling") as negative. It has been demonstrated that labeled words (or feature supervision) can greatly reduce the number of labeled samples for building high-quality classifiers [DMM08, ZE08]. In fact, different kinds of supervision generally have different acquisition costs, different degrees of utility and are not mutually redun-

dant [SML09]. Ideally, effective active learning schemes should be able to utilize different forms of supervision.

To incorporate the supervision on words and documents at same time into the active learning scheme, recently an active dual supervision (or dual active learning) has been proposed [MS09, SML09]. Comparing with traditional active learning, which aims to select the most "informative" examples (e.g., documents) for domain experts to label, active dual supervision selects both the "informative" examples (e.g., documents) and features (e.g., words) for labeling. For active dual supervision to be effective, there are three important components: a) an underlying learning mechanism that is able to learn from both the labeled examples and features (i.e., incorporating supervision on both examples and features); b) methods for estimating the value of information for example and feature labels; and c) a scheme that should be able to trade-off the costs and benefits of the different forms of supervision since they have different labeling costs and different benefits.

In the initial work on active dual supervision [SML09], a transductive bipartite graph regularization approach is used for learning from both labeled examples and features. In addition, uncertainty sampling and experimental design are used for selecting informative examples and features for labeling. To trade-off between different types of supervision, a simple probabilistic interleaving scheme where the active learner probabilistically queries the example oracle and the feature oracle is used. One problem in their method is that *the values of acquiring the feature labels and the example labels are not on the same scale*.

Recently, [LZS09] proposed a dual supervision method based on constrained non-negative tri-factorization of the document-term matrix where the labeled features and examples are naturally incorporated as sets of constraints. Having a framework for incorporating dual-supervision based on matrix factorization, gives rise to the natural question of *how to perform active dual supervision in this setting*. Since rows and columns are treated equally in estimating the errors of matrix factorization, another question is can we

make use of this characteristic of a matrix to address *the scaling issue in comparing the value of feature labels and example labels*.

In this chapter, we study the problem of active dual supervision using non-negative matrix tri-factorization. Our work is based on the dual supervision framework using constrained non-negative tri-factorization proposed in [LZS09]. We first extend the framework to explicitly model the corresponding relationships between feature classes and example classes. Then by making use of the reconstruction error criterion in matrix factorization, we propose a unified scheme to evaluate the value of feature and example labels. Instead of comparing the estimated performance increase of new feature labels or example labels, our proposed scheme assumes that a better supervision (a feature label or a example label) should lead to a more accurate reconstruction of the original data matrix. In our proposed scheme, *the value of feature labels and example labels is computed on the same scale*. The experiments show that our proposed unified scheme to query selection (i.e., feature/example selection for labeling) outperforms the interleaving schemes and the scheme based on expected log gain.

3.2 Related Work

Besides the literature of sentiment analysis discussed in 2.4, some previous research results that are most relevant to this work are highlighted in the following two directions: active learning and dual supervision.

3.2.1 Active Learning and Dual Active Learning

A recent report [Set09] surveys in depth on active learning. In this section, we briefly cover related work to position our contributions appropriately. Most prior work in active learning has focused on pooled-based techniques, where examples from an unlabeled pool

are selected for labeling [CAL94]. With the study of learning from labeled features, many research efforts on active learning with feature supervision are also reported [MSTPM05, RMJ06]. [GHSC04] proposed the notion of feature uncertainty and incorporated the acquired feature labels into learning by creating one-term mini-documents. [DSM09] performed active learning via feature labeling using several uncertainty reduction heuristics using the learning model developed in [DMM08]. [SML09] studied the problem of active dual supervision from examples and features using a graph-based dual supervision method with a simple probabilistic method for interleaving feature labels and example labels. In our work, we develop our active dual supervision framework using constrained non-negative tri-factorization and also propose a unified scheme to evaluate the value of feature and example labels. We note the very recent work of [AMP10], which proposes a unified approach for the dual active learning problem using expected utility where the utility is defined as the log gain of the classification model with a new labeled document or word. Conceptually, our proposed unified scheme is a special case of the expected utility framework where the utility is computed using the matrix reconstruction error. The utility based on the log gain of the classification model may not be reliable as small model changes resulted from a single additional example label or feature label may not be reflected in the classification performance [AMP10]. The empirical comparisons show that our proposed unified scheme based on reconstruction error outperforms the expected log gain.

3.2.2 Dual Supervision

Note that a learning method that is capable of performing dual supervision (i.e., learning from both labeled examples and features) is the basis for active dual supervision. Dual supervision is a relatively new area of research and few methods have been developed for

dual supervision. In [SM08, SHM09], a bipartite graph regularization model (GRADS) is used to diffuse label information along both sides of the document-term matrix and to perform dual supervision for semi-supervised sentiment analysis. Conceptually, their model implements a co-clustering assumption closely related to Singular Value Decomposition (see also [Dhi01, ZHD$^+$01] for more on this perspective). In [STUB08], standard regularization models are constrained using graphs of word co-occurrences. In [MGL09], Naive Bayes classifier is extended, where the parameters, the conditional word distributions given the classes, are estimated by combining multiple sources, e.g. document labels and word labels. Our work is based on the dual supervision framework using constrained non-negative tri-factorization.

3.3 Dual Supervision via Tri-NMF with Explicit Class Alignment

3.3.1 Learning with Dual Supervision via Tri-NMF

Our dual supervision model is based on non-negative matrix tri-factorization (Tri-NMF), where the non-negative input document-word matrix is approximated by 3 factor matrices as $X \approx GSF^T$, in which, X is an $n \times m$ document-term matrix, G is an $n \times k$ non-negative orthogonal matrix representing the probability of generating a document from a document cluster, F is an $m \times k$ non-negative orthogonal matrix representing the probability of generating a word from a word cluster, and S is a $k \times k$ nonnegative matrix providing the relationship between document cluster space and word cluster space.

While Tri-NMF is first applied in co-clustering, it is extended in [LZS09] to incorporate labeled words and documents as dual supervision via two loss terms in the objective

function of Tri-NMF as following:

$$\min_{F,G,S} \|X - GSF^T\|^2 + \alpha \operatorname{trace}[(F - F_0)^T C_1 (F - F_0)] + \beta \operatorname{trace}[(G - G_0)^T C_2 (G - G_0)]. \quad (3.1)$$

Here, $\alpha > 0$ is a parameter which determines the extent to which we enforce $F \approx F_0$ to its labeled rows. C_1 is a $m \times m$ diagonal matrix whose entry $(C_1)_{ii} = 1$ if the row of F_0 is labeled, that is, the class of the i-th word is known and $(C_1)_{ii} = 0$ otherwise. $\beta > 0$ is a parameter which determines the extent to which we enforce $G \approx G_0$ to its labeled rows. C_2 is a $n \times n$ diagonal matrix whose entry $(C_2)_{ii} = 1$ if the row of G_0 is labeled, that is, the category of the i-th document is known and $(C_2)_{ii} = 0$ otherwise. The squared loss terms ensure that the solution for G, F in the otherwise unsupervised learning problem be close to the prior knowledge G_0, F_0. So the partial labels on documents and words can be described using G_0 and F_0, respectively.

3.3.2 Modeling the Relationships between Word Classes and Document Classes

In the solution to Equation 3.1, we have $S = G^T X F$, or

$$S_{lk} = g_l^T X f_k = \frac{1}{|R_l|^{1/2} |C_k|^{1/2}} \sum_{i \in R_l} \sum_{j \in C_k} X_{ij}, \quad (3.2)$$

where $|R_l|$ is the size of the l-th document class, and $|C_k|$ is the size of the k-th word class [DLPP06]. Note that S_{lk} represents properly normalized within-class sum of weights ($l = k$) and between-class sum of weights ($l \neq k$). So, S represents the relationship between the classes over documents and the classes over words. Under the assumption that the i-th document class should correspond to the i-th word class, S should be an approximate diagonal matrix, since the documents of i-th class is more likely to contain the

words of the i-th class. Note that S is not an exact diagonal matrix, since a document of one class apparently can use words from other classes (especially G and F are required to be approximately orthogonal, which means the classification is rigorous). However, in Equation 3.1, there are no explicit constraints on the relationship between word classes and document classes. Instead, the relationship is established and enforced implicitly using existing labeled documents and words.

In active learning, the set of starting labeled documents or words is small, and this may generate an ill-formed S, leading to an incorrect alignment of word classes and document classes. To explicitly model the relationships between word classes and document classes, we constrain the shape of S via an extra loss term in the objective function as follows:

$$\begin{aligned}\min_{F,G,S} \quad & \|X - GSF^T\|^2 + \alpha \operatorname{trace}[(F - F_0)^T C_1 (F - F_0)] \\ & + \beta \operatorname{trace}[(G - G_0)^T C_2 (G - G_0)] + \gamma \operatorname{trace}[(S - S_0)^T (S - S_0)] \end{aligned} \quad (3.3)$$

where S_0 is a diagonal matrix. We will discuss the choice of S_0 in Section 3.3.4.

3.3.3 Computing Algorithm

This optimization problem can be solved using the following update rules

$$G_{jk} \leftarrow G_{jk} \frac{XFS + \beta C_2 G_0}{(GG^T XFS + \beta GG^T C_2 G)_{jk}}, \quad (3.4)$$

$$S_{jk} \leftarrow S_{jk} \frac{F^T X^T G + \gamma S_0}{(F^T FSG^T G + \gamma S)_{jk}}, \quad (3.5)$$

$$F_{jk} \leftarrow F_{jk} \frac{X^T GS^T + \alpha C_1 F_0}{(FF^T X^T GS^T + \alpha C_1 F)_{jk}}. \quad (3.6)$$

The algorithm consists of an iterative procedure using the above three rules until convergence.

Theorem 3.3.1 *The solution satisfies the Karuch-Kuhn-Tucker (KKT) optimality condition, i.e., the algorithm converges correctly to a local optima.*

Proof. Proof of the updates of F and G is the same as in [LZS09]. Here we focus on the update rule of S. We want to minimize

$$\mathcal{L}(S) = \|X - GSF^T\| + \alpha \operatorname{trace}[(F - F_0)^T C_1 (F - F_0)] + \beta \operatorname{trace}[(G - G_0)^T C_2 (G - G_0)] + \gamma \operatorname{trace}[(S - S_0)^T (S - S_0)]. \quad (3.7)$$

The gradient of L is

$$\frac{\partial \mathcal{L}}{\partial S} = 2F^T FSG^T G - 2F^T X^T G + 2\gamma(S - S_0)$$

The KKT complementarity condition for the non-negativity of S_{jk} gives

$$[2F^T FSG^T G - 2F^T X^T G + 2\gamma(S - S_0)]_{jk} S_{jk} = 0.$$

This is the fixed point relation that local minima for S must satisfy, which is equivalent with the update rule of S in Equation 3.6. □

3.3.4 Probabilistic Interpretation of Tri-NMF

If X is L_1 normalized, then the entries of X present the joint probability distribution of word and document $p(d, w)$, which can be decomposed as follows:

$$p(d, w) = \sum p(d, w|z_d, z_w) p(z_d, z_w), \quad (3.8)$$

$$= \sum p(w|z_w) p(d|z_d) p(z_d, z_w), \quad (3.9)$$

where we have used the conditional independence $p(d, w|z_d, z_w) = p(w|z_w)p(d|z_d)$. Here random variables w,d represent the word and document respectively, and z_w, z_d are latent class variables.

If we set

$$F_{il} = p(w = w_i|z_w = l), \quad (3.10)$$

$$G_{jk} = p(d = d_j|z_d = k), \quad (3.11)$$

$$S_{kl} = p(z_d = k, z_w = l), \quad (3.12)$$

then

$$(GSF^T)_{ij} = \sum_{k,l} G_{d=d_i,z_d=k} S_{z_d=k,z_w=l} F_{w=w_j,z_w=k} \tag{3.13}$$

$$= \sum_{k,l} [p(d = d_i|z_d = k)p(w = w_j|z_w = l)p(z_d = k, z_w = l)] \tag{3.14}$$

$$= p(d = d_i, w = w_j). \tag{3.15}$$

So if X is L_1 normalized, the 3 factors G, F, S of Tri-NMF can be interpreted as the conditional document distributions given the document class, conditional word distributions given the word class, and the joint distribution of a document class and a word class. Given K word/document classes, according to the probability interpretation, we can estimate S_0 as follows:

$$[S_0]_{kl} = p(z_d = k, z_w = k) \tag{3.16}$$

$$= \begin{cases} 1/K & l = k, \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

3.4 A Unified Query Selection Scheme Using Reconstruction Error

An ideal active dual supervision scheme should be able to evaluate the value of acquiring labels for documents and words on the same scale. In the initial study of dual active supervision, different scores are used for documents and words (e.g. uncertainty for documents and certainty for words), and thus they are not on the same scale [SML09]. Recently, the framework of Expected Utility (Estimated Risk Minimization) is proposed in [AMP10]. At each step of the framework, the next word or document selected for labeling is the one that will result in the highest estimated improvement in classifier performance as defined as:

$$EU(q_j) = \sum_{k=1}^{K} P(q_j = c_k)U(q_j = c_k), \tag{3.18}$$

where K is the class number, $P(q_j = c_k)$ indicates the probability that q_j, j-th query (a word or document), belongs to the k-th class, and the $U(q_j = c_k)$ indicates the utility that q_j belongs to the k-th class. However, the choice of the utility measure is still a challenge.

3.4.1 Reconstruction Error

In our matrix factorization framework, rows and columns are treated equally in estimating the errors of matrix factorization, and the reconstruction error is thus a natural measure of utility. Let the current supervision knowledge be G_0, F_0. To select a new unlabeled document/word for labeling, we assume that a good supervision should lead to a good constrained factorization for the document-term matrix, $X \approx GSF^T$. If the new query q_j is a word and its label is k, then the new factorization is

$$G^*_{j=k}, S^*_{j=k}, F^*_{j=k} = \arg\min_{G,S,F} \|X - GSF^T\|^2 \alpha \operatorname{trace}[(G - G_0)^T C_2 (G - G_0)] + \beta \operatorname{trace}[(F - F_{0,j=k})^T C_1 (F - F_{0,j=k})] + \gamma \operatorname{trace}[(S - S_0)^T (S - S_0)], \quad (3.19)$$

where $F_{0,j=k}$ is same as F_0 except that $F_{0,j=k}(j,k) = 1$. In other words, we obtained a new factorization using the labeled words. Similarly, if the new query q_j is a document, then the new factorization is

$$G^*_{j=k}, S^*_{j=k}, F^*_{j=k} = \arg\min_{G,S,F} \|X - GSF^T\|^2 + \alpha \operatorname{trace}[(G - G_{0,j=k})^T C_2 (G - G_{0,j=k})] + \beta \operatorname{trace}[(F - F_0)^T C_1 (F - F_0)] + \gamma \operatorname{trace}[(S - S_0)^T (S - S_0)], \quad (3.20)$$

where $G_{0,j=k}$ is same as G_0 except that $G_{0,j=k}(j,k) = 1$. In other words, we obtained a new factorization using the labeled documents. Then the new reconstruction error is

$$RE(q_j = k) = \|X - G^*_{j=k} S^*_{j=k} F^*_{j=k}\|^2. \quad (3.21)$$

So the expected utility of a document or word label query, q_j, can be computed as

$$EU(q_j) = \sum_{k=1}^{K} P(q_j = k) * (-RE(q_j = k)). \tag{3.22}$$

3.4.2 Algorithm Description

Computational Improvement: It can be computationally intensive if the reconstruction error is computed for all unknown documents and words. Inspired by [AMP10], we first select the top 100 unknown words that the current model is most certain about, and the top 100 unknown documents that the current model is most uncertain about. Then we identify the words or documents in this pool with the highest expected utility (reconstruction error). As discussed in Section 3.3.4, the posterior distribution for words and documents can be estimated using the factors of Tri-NMF as follows:

$$p(z_w = k|w = w_i) \propto p(w = w_i|z_w = k) \sum_{j=1}^{K} p(z_w = k, z_d = j) \tag{3.23}$$

$$= F_{ik} * \sum_{j=1}^{K} S_{kj}. \tag{3.24}$$

$$p(z_d = k|d = d_i) \propto p(d = d_i|z_d = k) \sum_{j=1}^{K} p(z_w = j, z_d = k) \tag{3.25}$$

$$= G_{ik} * \sum_{j=1}^{K} S_{jk}. \tag{3.26}$$

Thus, Equations 3.23 and 3.25 are used to perform the initial selection of top 100 unknown words and top 100 unknown documents.

The overall algorithm procedure is described in Algorithm 1. First we iteratively use the updating rules of Equation 3.6 to obtain the factorization G, F, S based on initial labeled documents and words. Then to select a new query, for each unlabeled document or word in the pool and for each possible class, we compute the reconstruction error with new

Algorithm 1 Active Dual Supervision Algorithm Based on Matrix Factorization

INPUT: X, document-word matrix; F_0, current labeled words; G_0, current labeled documents; O, the oracle

OUTPUT: G, classification result for all documents in X

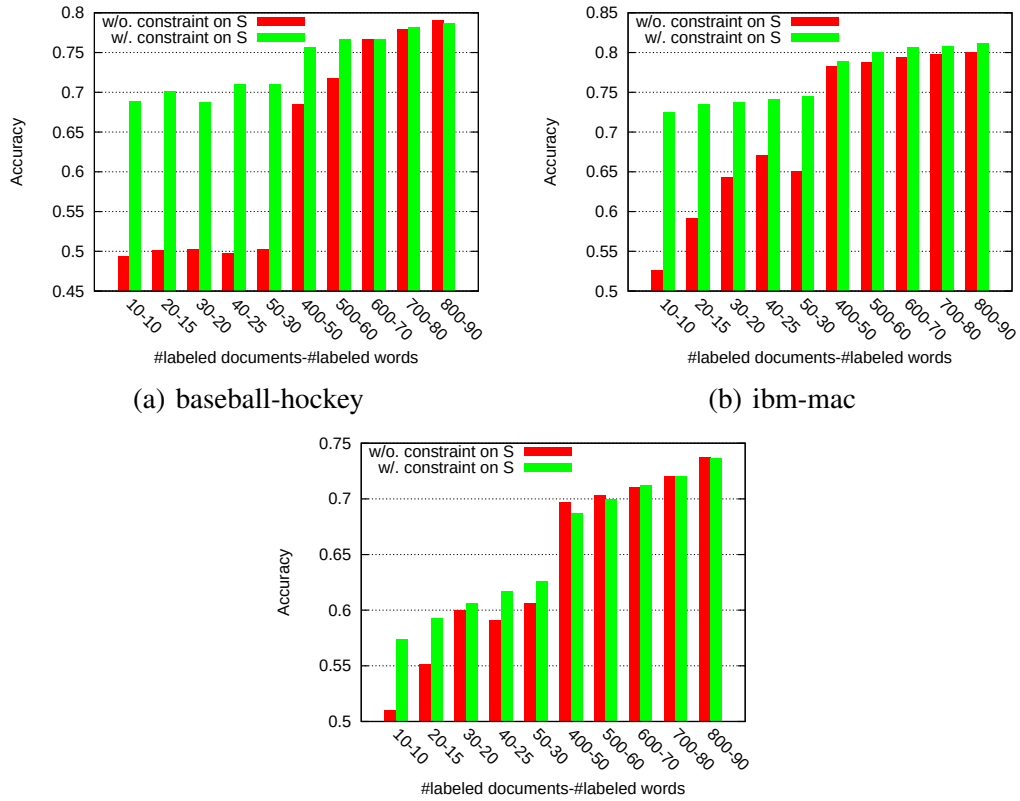
1. Get base factorization of X: G, S, F.
2. Active dual supervision

repeat
  D is the set of top 100 unlabeled documents with most uncertainty;
  W is the set of top 100 unlabeled words with most certainty;
  $Q = D \cup W$;
  for all $q \in Q$ **do**
    for $k = 1$ to K **do**
      Get $G^*_{q=k}, F^*_{q=k}, S^*_{q=k}$ by Equation 3.19 or Equation 3.20 according to whether the query q is a document or a word;
    Calculate $EU(q)$ by Equation 3.22;
  $q* = \arg\max_q EU(q)$;
  Acquire new label of q^*, l from O;
  $G, F, S = G^*_{q^*=l}, F^*_{q^*=l}, S^*_{q^*=l}$;
until stop criterion is met.

supervision (using the current factorization results as initialization values). It is efficient to compute a new factorization due to the sparsity of the matrices. The document-term matrix is typically very sparse with $z \ll nm$ non-zero entries while k is typically also much smaller than document number n, and word number m. By using sparse matrix multiplications and avoiding dense intermediate matrices, updating F, S, G each takes $O(k^2(m + n) + kz)$ time per iteration which scales linearly with the dimensions and density of the data matrix [LZS09]. Empirically, the number of iterations that is needed to compute the new factorization is usually very small (less than 10).

3.5 Experiments

We conduct our experiments on both topic classification and sentiment analysis tasks.



(a) baseball-hockey

(b) ibm-mac

(c) med-space

Figure 3.1: Comparing the performance of dual supervision via Tri-NMF w/ and w/o the constraint on S.

3.5.1 Topic Classification

Three popular binary text classification datasets are used in the experiments: ibm-mac (1937 examples), baseball-hockey (1988 examples) and med-space (1972 examples) datasets. All of them are drawn from the 20-newsgroups text collection[1] where the task is to assign messages into the newsgroup in which they appeared. Top 1500 frequent words in each dataset are used as features in the binary vector representation. These datasets have labels for all the documents. For a document query, the oracle returns its label. We construct the word oracle in the same manner as in [SML09]: first compute the information gain of words with respect to the known true class labels in the training splits of a dataset,

[1] http://www.ai.mit.edu/people/jrennie/20_newsgroups/

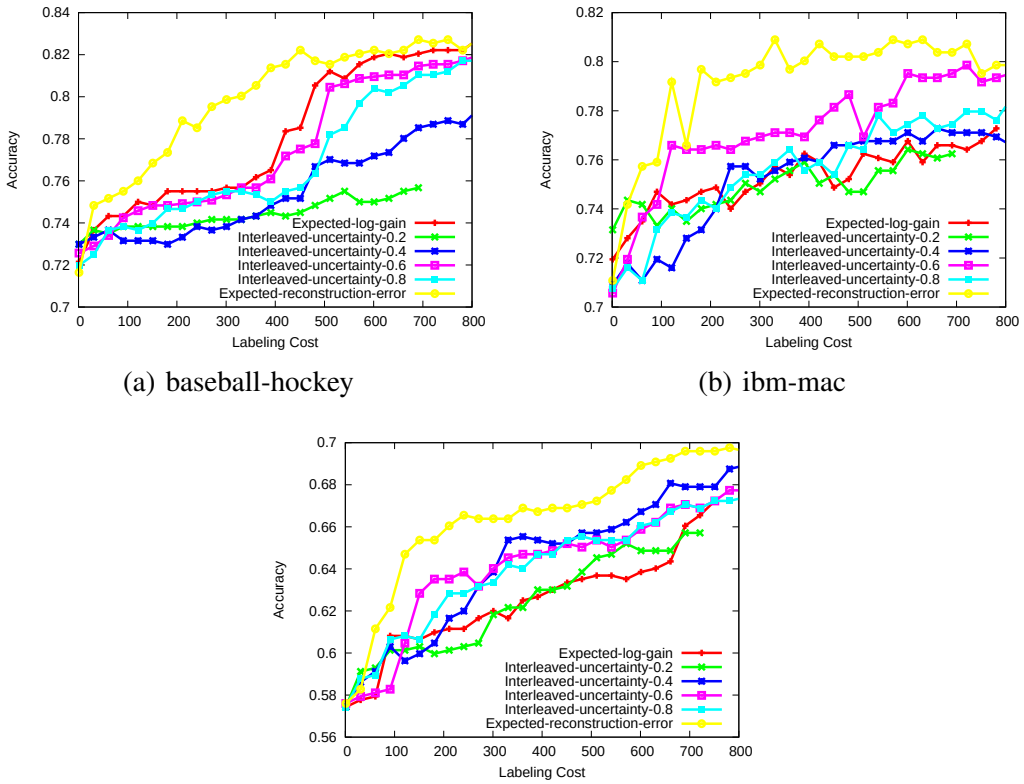
and then the top 100 words as ranked by information gain are assigned the label which is the class in which the word appears more frequently. To those words with labels, the word oracle returns its label; otherwise, the oracle returns a "don't know" response (no word label is obtained for learning, but the word is excluded from the following query selection).

Results are averaged over 10 random training-test splits. For each split, 30% examples are used for testing. All methods are initialized by a random choice of 10 document labels and 10 word labels. For simplicity, we follow the widely used cost model [RA07, DMM08, SML09] where features are roughly 5 times cheaper to label than examples, so we assume the cost is 1 for a word query and is 5 for a document query. We set $\alpha = \beta = 5$, $\gamma = 1$ for all the following experiments[2].

Effect of Constraints on S in Constrained Tri-NMF Figure 3.1 demonstrates the effectiveness of dual supervision with explicit class alignment via Tri-NMF as described in Section 3.3. When there are enough labeled documents and words, the constraints on S have a relative small impact on the performance of dual supervision. However, in the beginning phase of active learning, the labeled dataset can be small (such as 10 labeled documents and 10 labeled words). In this case, without the constraint of S, the matrix factorization may generate incorrect class alignment, thus lead to almost random classification results (around 50% accuracy), as shown in Figure 1, and further make unreasonable the following evaluation of queries.

Comparing Query Selection Approaches Figure 3.2 compares our proposed unified scheme (denoted as *Expected-reconstruction-error*) with the following baselines using

[2]We do not perform fine tuning on the parameters since the main objective of the paper is to demonstrate the effectiveness of matrix factorization based methods for dual active supervision. A vigorous investigation on the parameter choices is our further work.



(a) baseball-hockey

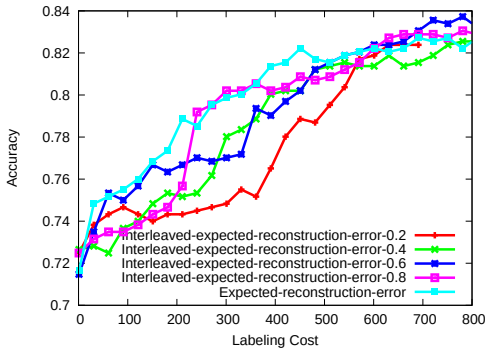
(b) ibm-mac

(c) med-space

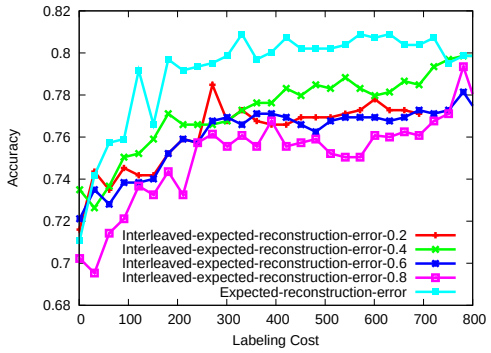
Figure 3.2: Comparing the different query selection approaches in active learning via Tri-NMF with dual supervision.

Tri-NMF as the classifier for dual supervision: (1). *Interleaved-uncertainty* which first selects feature query by certainty and sample query by uncertainty and then combines the two types of queries using an interleaving scheme. The interleaving probability (probability to select the query as a document) is set as 0.2, 0.4, 0.6 and 0.8. (2). *Expected-log-gain* which selects feature and sample query by maximizing the expected log gain. *Expected-reconstruction-error* outperforms interleaving schemes with all the different interleaving probability values with which we experimented. It also has a better performance than *Expected-log-gain*. Although log gain is a finer-grained utility measure of classifier performance than accuracy and has a good performance in the setting with a large set of starting labeled documents (e.g., 100 documents), it is not reliable especially in the setting with a small set of labeled data. Different from the *Expected-log-gain*, *Expected-reconstruction-error* estimates the utility using the matrix reconstruction error, making use of information of all documents and words, including those unlabeled.

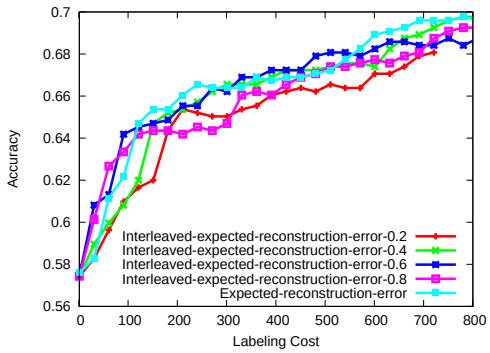
Interleaving Scheme vs. the Unified Scheme To further demonstrate the benefit of the proposed unified scheme, we compare it with its interleaved version: *Interleaved-expected-construction-error* which computes the utility of a query using the reconstruction error, but uses interleaving scheme to decide which type of query to select. We experiment with different interleaving probability values ranging from 0.2 to 0.8, which lead to quite different performance results. From Figure 3.3, the optimal interleaving probability value varies on different datasets. For example, the probability value of 0.8 is among the optimal interleaving probability values on baseball-hockey dataset but performs poorly on ibm-mac dataset. This observation also illustrates the need for a unified scheme, because of the difficulty in choosing the optimal interleaving probability value. Although the proposed unified scheme is not significantly better than its interleaving counterparts for all interleaving probability values on all datasets, it avoids bad choices.



(a) baseball-hockey

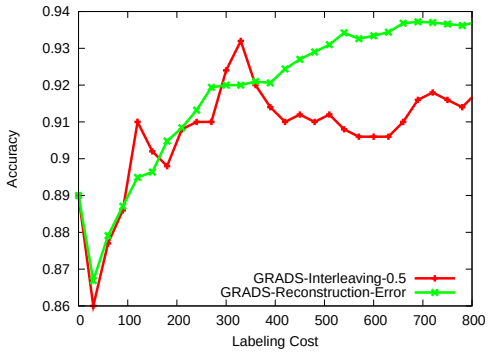


(b) ibm-mac

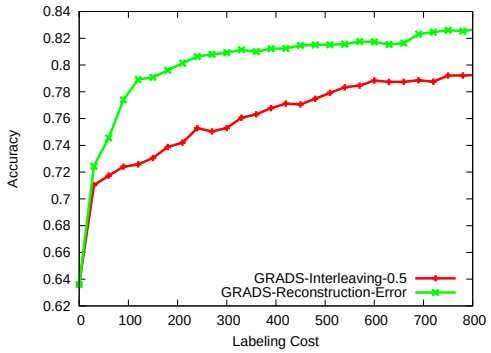


(c) med-space

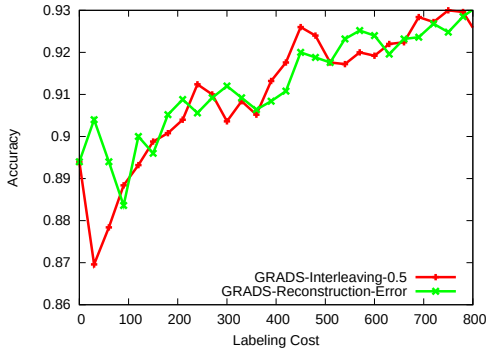
Figure 3.3: Comparing the unified and interleaving scheme based on reconstruction error.



(a) baseball-hockey

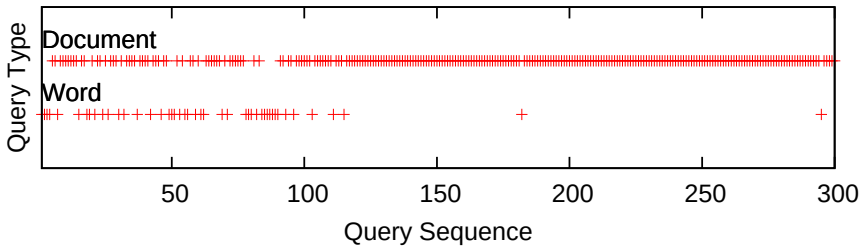


(b) ibm-mac

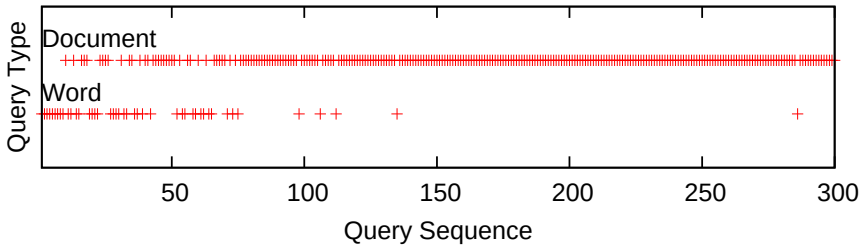


(c) med-space

Figure 3.4: GRADS with reconstruction error and interleaving uncertainty.



(a) baseball-hockey



(b) ibm-mac

Figure 3.5: Example of query sequence.

Figure 3.5 presents the sequence of different query types selected by our unified scheme and it clearly demonstrates the distribution patterns of different query types. At the beginning phase of active learning, word queries have much higher probabilities to be selected, which is consistent with the result of previous work: feature labels can be more effective than examples in text classification [DMM08]. And in the later learning phase, documents are more likely to be selected, since the number of words that can benefit the classification is much smaller than the effective documents.

Reconstruction Error vs. Interleaving uncertainty using GRADS It should be pointed out that *our unified scheme for query selection based on reconstruction error does not rely on the estimation of model performance on training data and can be easily integrated with other dual supervision models* such as GRADS [SHM09]. Figure 3.4 shows the comparison of GRADS using the interleaved scheme with an interleaving probability of 0.5, and using our unified scheme based on reconstruction error. Among the 3 datasets

we used, the reconstruction error based approach outperforms the interleaving scheme on baseball-hockey and ibm-mac, and has similar performance with the interleaving scheme on med-space.

3.5.2 Sentiment Classification

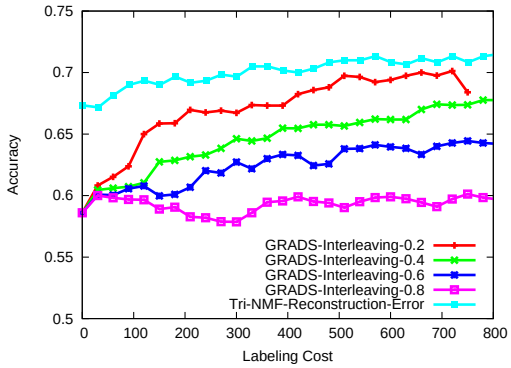


Figure 3.6: Comparing active dual supervision using matrix factorization with GRADS on sentiment analysis.

We also comparing active dual supervision using matrix factorization with GRADS on the sentiment classification task. The sentiment analysis experiment is conducted on the movies review dataset [PLV02], containing 1000 positive and 1000 negative movie reviews. The results are shown in Figure 3.6. The experimental results clearly demonstrate the effectiveness of our approach, denoted as *Tri-NMF-Reconstruction-Error*.

3.6 Summary

In this chapter, we study the problem of dual active supervision, and propose a matrix tri-factorization based approach to address how to evaluate labeling benefit of different types of queries (examples or features) in the same scale. We first extend the nonnegative matrix tri-factorization to the dual active supervision setting, and then use the reconstruction

error to evaluate the value of feature and example labels. Experimental results show that our proposed approach outperforms existing methods in both topic classification and sentiment classification.

CHAPTER 4

PARTICIPANT-BASED EVENT DETECTION ON TWITTER STREAMS

4.1 Introduction

Twitter, one of the most representative examples of micro-blogging service providers, allows users to post short messages, *tweets*, within 140-character limit. One particular topic Twitter users publish tweets about is "what's happening", which makes Twitter differentiated from news media with its real-time nature. For example, we could detect a tweet related to a shooting crime 10 minutes after shots fired, while the first new report appeared approximately three hours later. Meanwhile, tweets have a broad coverage over all types of real-world events, accounting for Twitter's large number of users, including verified accounts such as news agents, organizations and public figures. The real-time event information is particularly useful for keep people informed and updated on the events happening in real-world with their user-contributed messages.

Although the large volume of tweets provides enough information about events, because of a lot of noises, it is not straightforward and sometimes difficult for people themself to access the real information about a particular event from the Twitter stream. To make use of Twitter's real-time nature, it is imperative to develop effective automatic methods to conduct event detection, detecting events from a Twitter stream by identifying important moments in the stream and their associated tweets.

Most of existing approaches[ZZW+12, MBB+11, WL11, ZSAG12] rely on changes of tweet volumes by detecting bursts in the stream as important moments, and assume all tweets during a burst describe the corresponding event. However in real cases, because of average effects of multiple topics existing in the stream, important moments, in term of one topic, which may lead bursts among posts about the topic, may not be well reflected in changes of post volumes in the whole stream. This can be shown using an example in

Figure 4.1: Example Twitter event stream (upper) and participant stream (lower).

Figure 4.1, in which upper one is a Twitter stream which is composed of tweets related to a NBA game Spurs vs Thunder, and the lower one is its sub-stream which contains only tweets corresponding to the player Russell Westbrook in this game.

Previous research on event detection focuses on identifying the important moments from the coarse-level event stream. This may yield several side effects: first, the spike patterns are not clearly identifiable from the overall event stream, though they are more clearly seen if we "zoom-in" to the participant level; second, it is arguable whether the important events can be accurately detected based solely on the tweet volume change; third, a popular participant or event can elicit huge volume of tweets which dominant the entire stream discussion and shield less prominent events. For example, in the NBA games, discussions about the key players (e.g., "LeBron James", "Kobe Bryant") can heavily shadow other important participants or events, resulting in detected event list with repetitive events about the dominant players.

In this chapter, we propose a novel participant-based event detection approach, which dynamically identifies the participants from data streams, and then "zooms-in" the twitter stream to participant level to detect the important events related to each participant using a novel time-content mixture model. Results show that the mixture model-based event detection approach can efficiently incorporate the "burstiness" and "cohesiveness" of the participant streams, and the participant-based event detection can effectively capture the events that have otherwise been shadowed by the long-tail of other dominant events, yielding final result with considerably better coverage than the state-of-the-art approach.

4.2 Participant-based Event Detection

We propose a novel participant-centered event detection approach that consists of two key components: (1) "Participant Detection" dynamically identifies the event participants and divides the entire stream into a number of participant streams (Section 4.2.1); (2) "Event Detection" introduces a novel time-content mixture model approach (Section 4.2.2) to identify the important events associated with each participant; these "participant-level events" are then merged along the timeline to form a set of "global events"[1], which capture all the important moments in the given stream.

4.2.1 Participant Detection

We define event participants as the entities that play a significant role in the event. "Participant" is a general concept to denote the event participating persons, organizations,

[1]We use "**participant events**" and "**global events**" respectively to represent the important moments happened on the participant-level and on the entire event-level. A "global event" may consist of one or more "participant events". For example., the "steal" action in the basketball game typically involves both the defensive and offensive players, and can be generated by merging the two participant-level events.

product lines, etc., each of which can be captured by a set of correlated proper nouns. For example, the NBA player "*LeBron Raymone James*" can be represented by {*LeBron James, LeBron, LBJ, King James, L. James*}, where each proper noun represents a unique mention of the participant. In this work, we automatically identify the proper nouns from tweet streams, filter out the infrequent ones using a threshold ψ, and cluster them into individual event participants. This process allows us to dynamically identify the key participating entities and provide a full-coverage for these participants in the detected events.

We formulate the participant detection in a hierarchical agglomerative clustering framework. The CMU TweetNLP tool [GSO+11] was used for proper noun tagging. The proper nouns (a.k.a., mentions) are grouped into clusters in a bottom-up fashion. Two mentions are considered similar if they share (1) lexical resemblance, and (2) contextual similarity. For example, in the following two tweets "*Gotta respect Anthony Davis, still rocking the unibrow*", "*Anthony gotta do something about that unibrow*", the two mentions *Anthony Davis* and *Anthony* are referring to the same participant and they share both character overlap ("anthony") and context words ("unibrow", "gotta"). We use $sim(c_i, c_j)$ to represent the similarity between two mentions c_i and c_j, defined as:

$$sim(c_i, c_j) = lex\_sim(c_i, c_j) \times cont\_sim(c_i, c_j)$$

where the lexical similarity ($lex\_sim(\cdot)$) is defined as a binary function representing whether a mention c_i is an abbreviation, acronym, or part of another mention c_j, or if the character edit distance between the two mentions is less than a threshold θ[2]:

$$lex\_sim(c_i, c_j) = \begin{cases} 1 & c_i(c_j) \text{ is part of } c_j(c_i) \\ 1 & \text{EditDist}(c_i, c_j) < \theta \\ 0 & \text{Otherwise} \end{cases}$$

[2] θ was empirically set as $0.2 \times min\{|c_i|, |c_j|\}$

We define the context similarity ($cont\_sim(\cdot)$) of two mentions as the cosine similarity between their context vectors $\vec{v}_i$ and $\vec{v}_j$. Note that on the tweet stream, two temporally distant tweets can be very different even though they are lexically similar, e.g., two slam dunk shots performed by the same player at different time points are different. We therefore restrain the context to a segment of the tweet stream $|S_k|$ and then take the weighted average of the segment-based similarity as the final context similarity. To build the context vector, we use term frequency (TF) as the term weight and remove all the stop-words. We use $|D|$ to represent the total tweets in the event stream.

$$cont\_sim_{|S_k|}(c_i, c_j) = \cos(\vec{v}_i, \vec{v}_j)$$

$$cont\_sim(c_i, c_j) = \sum_k \frac{|S_k|}{|D|} \times cont\_sim_{|S_k|}(c_i, c_j)$$

Similarity between two clusters of mentions are defined as the maximum possible similarity between a pair of mentions, each from one cluster:

$$sim(C_i, C_j) = \max_{c_i \in C_i, c_j \in C_j} sim(c_i, c_j)$$

We perform bottom-up agglomerative clustering on the mentions until a stopping threshold δ has been reached for $sim(C_i, C_j)$. The clustering approach naturally groups the frequent proper nouns into participants. The **participant streams** are then formed by gathering the tweets that contain one or more mentions in the participant cluster.

4.2.2 Mixture Model-based Event Detection

An event corresponds to a topic that emerges from the data stream, being intensively discussed during a time period, and then gradually fades away. The tweets corresponding to an event thus demand not only "temporal burstiness" but also a certain degree of "lexical cohesiveness". To incorporate both the time and content aspects of the events, we propose a mixture model approach for event detection. Figure 4.2 shows the plate notation.

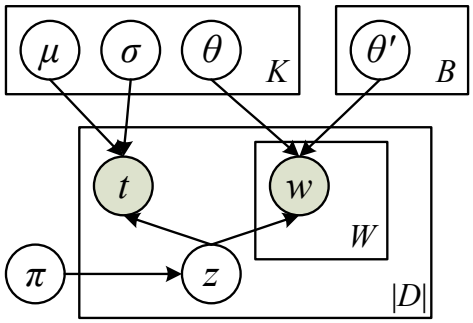


Figure 4.2: Plate notation of the mixture model.

In the proposed model, each tweet d in the data stream D is generated from a topic z, weighted by π_z. Each topic is characterized by both its content and time aspects. The content aspect is captured by a multinomial distribution over the words, parameterized by θ; while the time aspect is characterized by a Gaussian distribution, parameterized by μ and σ, with μ represents the average time point that the event emerges and σ determines the duration of the event. These distributions bear similarities with the previous work [Hof99, All02, HV09]. In addition, there are often background or "noise" topics that are being constantly discussed over the entire event evolvement process and do not present the desired "burstiness" property. We use a uniform distribution $U(t_b, t_e)$ to model the time aspect of these "background" topics, with t_b and t_e being the event beginning and end time points. The content aspect of a background topic is modeled by similar multinomial distribution, parameterized by θ'. We use the maximum likelihood parameter estimation. The data likelihood can be represented as:

$$L(D) = \prod_{d \in D} \sum_{z} \{\pi_z p_z(t_d) \prod_{w \in d} p_z(w)\}$$

where $p_z(t_d)$ models the timestamp of tweet d under the topic z; $p_z(w)$ corresponds to the word distribution in topic z. They are defined as:

$$p_z(t_d) = \begin{cases} N(t_d; \mu_z, \sigma_z) & \text{if } z \text{ is an event topic} \\ U(t_b, t_e) & \text{if } z \text{ is background topic} \end{cases}$$

$$p_z(w) = \begin{cases} p(w;\theta_z) & \text{if } z \text{ is an event topic} \\ p(w;\theta'_z) & \text{if } z \text{ is background topic} \end{cases}$$

where both $p(w;\theta_z)$ and $p(w;\theta'_z)$ are multinomial distributions over the words. Initially, we assume there are K event topics and B background topics and use the EM algorithm for model fitting. The EM equations are listed below:

E-step:

$$p(z_d = j) \propto \begin{cases} \pi_j N(d;\mu_j,\sigma_j) \prod_{w\in d} p(w;\theta_j) & \text{if } j <= K \\ \pi_j U(t_b,t_e) \prod_{w\in d} p(w;\theta'_j) & \text{else} \end{cases}$$

M-step:

$$\pi_j \propto \sum_d p(z_d = j)$$

$$p(w;\theta_j) \propto \sum_d p(z_d = j) \times c(w,d)$$

$$p(w;\theta'_j) \propto \sum_d p(z_d = j) \times c(w,d)$$

$$\mu_j = \frac{\sum_d p(z_d = j) \times t_d}{\sum_{j=1}^K \sum_d p(z_d = j)}$$

$$\sigma_j^2 = \frac{\sum_d p(z_d = j) \times (t_d - \mu_j)^2}{\sum_{j=1}^K \sum_d p(z_d = j)}$$

To process the data stream D, we divide the data into 10-second bins and process each bin at a time. The peak time of an event was determined as the bin that has the most tweets related to this event. During EM initialization, the number of event topics K was empirically decided by scanning through the data stream and examine tweets in every 3-minute stream segment. If there was a spike[3], we add a new event to the model

[3]We use the algorithm described in [MBB+11] as a baseline and ad hoc spike detection algorithm.

and use the tweets in this segment to initialize the value of μ, σ, and θ. Initially, we use a fixed number of background topics with $B = 4$. A topic re-adjustment was performed after the EM process. We merge two events in a data stream if they (1) locate closely in the timeline, with peaks times within a 2-minute window; and (2) share similar word distributions: among the top-10 words with highest probability in the word distributions, there are over 5 words overlap. We also convert the event topics to background topics if their σ values are greater than a threshold β[4]. We then re-run the EM process to obtain the updated parameters. The topic re-adjustment process continues until the number of events and background topics do not change further.

We obtain the "**participant events**" by applying this event detection approach to each of the participant streams. The "**global events**" are obtained by merging the participant events along the timeline. We merge two participant events into a global event if (1) their peaks are within a 2-minute window, and (2) the Jaccard similarity [L.99] between their associated tweets is greater than a threshold (set to 0.1 empirically). The tweets associated with each global event are the ones with $p(z|d)$ greater than a threshold γ, where z is one of the participant events and γ was set to 0.7 empirically. After the event detection process, we obtain a set of global events and their associated event tweets.[5]

4.3 Experiments

4.3.1 Experimental Data

We evaluate the proposed event detection approach on seven datasets: six NBA basketball games and a conference speech, namely the Apple CEO's keynote speech in the Apple

[4] β was set to 5 minutes in our experiments.

[5] We empirically set some threshold values in the topic re-adjustment and event merging process. In future, we would like to explore more principled way of parameter selection.

Event		Date	Duration	#Tweets
	Lakers vs Okc	05/19/2012	3h10m	218,313
N	Celtics vs 76ers	05/23/2012	3h30m	245,734
B	Celtics vs Heat	05/30/2012	3h30m	345,335
A	Spurs vs Okc	05/31/2012	3h	254,670
	Heat vs Okc (1)	06/12/2012	3h30m	331,498
	Heat vs Okc (2)	06/21/2012	3h30m	332,223
Apple's WWDC'12 Conf.		06/11/2012	3h30m	163,775

Table 4.1: Statistics of the data set, including six NBA basketball games and the WWDC 2012 conference event.

Worldwide Developers Conference (WWDC 2012)[6]. Althought each of the datasets itself can be seen corresponding to an event (referred to as an event topic in the following), our goal is to detect finer-grained events, which are easier to evaluate.

We use the heterogeneous event topics to verify that the proposed approach can robustly and efficiently detect events on different types of Twitter streams. The tweet streams corresponding to these topics are collected using the Twitter Streaming API[7] with pre-defined keyword set. For NBA games, we use the team names, first name and last name of the players and head coaches as keywords for retrieving the tweets related to the event topic; for the WWDC conference, the keyword set contains about 20 terms related to Apple, such as "wwdc", "apple", "mac", etc. We crawl the tweets in real-time when these scheduled events are taking place; nevertheless, certain non-event tweets could be mis-included due to the broad coverage of the used keywords. During preprocessing, we filter out the tweets containing URLs, non-English tweets, and retweets since they are less likely containing new information regarding the event progress. Table 4.1 shows statistics of the event tweets after the filtering process. In total, there are over 1.8 million tweets used in the event detection experiments.

[6]https://developer.apple.com/wwdc/

[7]https://dev.twitter.com/docs/streaming-apis

Time	Action (Event)	Score
9:22	Chris Bosh misses 10-foot two point shot	7-2
9:22	Serge Ibaka defensive rebound	7-2
9:11	Kevin Durant makes 15-foot two point shot	9-2
8:55	Serge Ibaka shooting foul (Shane Battier draws the foul)	9-2
8:55	Shane Battier misses free throw 1 of 2	9-2
8:55	Miami offensive team rebound	9-2
8:55	Shane Battier makes free throw 2 of 2	9-3

Table 4.2: An example clip of the play-by-play live coverage of an NBA game (Heat vs Okc).

We use the play-by-play live coverage collected from the ESPN[8] and MacRumors[9] websites as reference, which provide detailed descriptions of the NBA and WWDC as they unfold. Table 4.2 shows an example clip of the play-by-play descriptions of an NBA game, where "Time" corresponds to the minutes left in the current quarter of the game, and "Score" shows the score between the two teams. Ideally, each item in the live coverage descriptions may correspond to an event in the tweet streams, but in reality, not all actions would attract enough attention from the Twitter audience. We use a human annotator to manually filter out the actions that did not lead to any spike in the corresponding participant stream. The rest items are projected to the participant and event streams as the goldstandard events. The projection was manually performed since the "game clock" associated with the goldstandard (first column in Table 4.2) does not align well with the "wall clock" due to the game rules such as timeout and halftime rest. To evaluate the participant detection performance, we ask the annotator to manually group the proper noun mentions into clusters, each cluster corresponds to a participant. The mentions that do not correspond to any participant are discarded.

[8]http://espn.go.com/nba/scoreboard

[9]http://www.macrumorslive.com/archive/wwdc12/

Example Participants - NBA game
westbrook, russell westbrook
stephen jackson, steven jackson, jackson
james, james harden, harden
ibaka, serge ibaka
oklahoma city thunder, oklahoma
gregg popovich, greg popovich, popovich
kevin durant, kd, durant
thunder, okc, #okc, okc thunder, #thunder
Example Participants - WWDC Conference
macbooks, mbp, macbook pro, macbook air,...
google maps, google, apple maps
wwdc, apple wwdc, #wwdc
os, mountain, os x mountain, os x
iphone 4s, iphone 3gs, iphone

Table 4.3: Example participants automatically detected from the NBA game Spurs vs Okc (2012-5-31) and the WWDC'12 conference.

4.3.2 Participant Detection Results

In Table 4.3, we show example participants that were automatically detected by the proposed hierarchical agglomerative clustering approach. We note that the clusters include various mentions of the same event participant, e.g., "*gregg popovich*", "*greg popovich*", and "*popovich*" are both referring to the head coach of the team Spurs; "*macbooks*", "*macbook pro*", "*mbp*" are referring to a line of products from Apple. Quantitatively, we evaluate the participant detection results on both participant- and mention-level. Assume the system-detected and the goldstandard participant clusters are T_s and T_g respectively. We define a **correct participant** as a system detected participant with more than half of its associated mentions are included in a goldstandard participant (referred to as the **hit participant**). As a result, we can define the participant-level precision and recall as

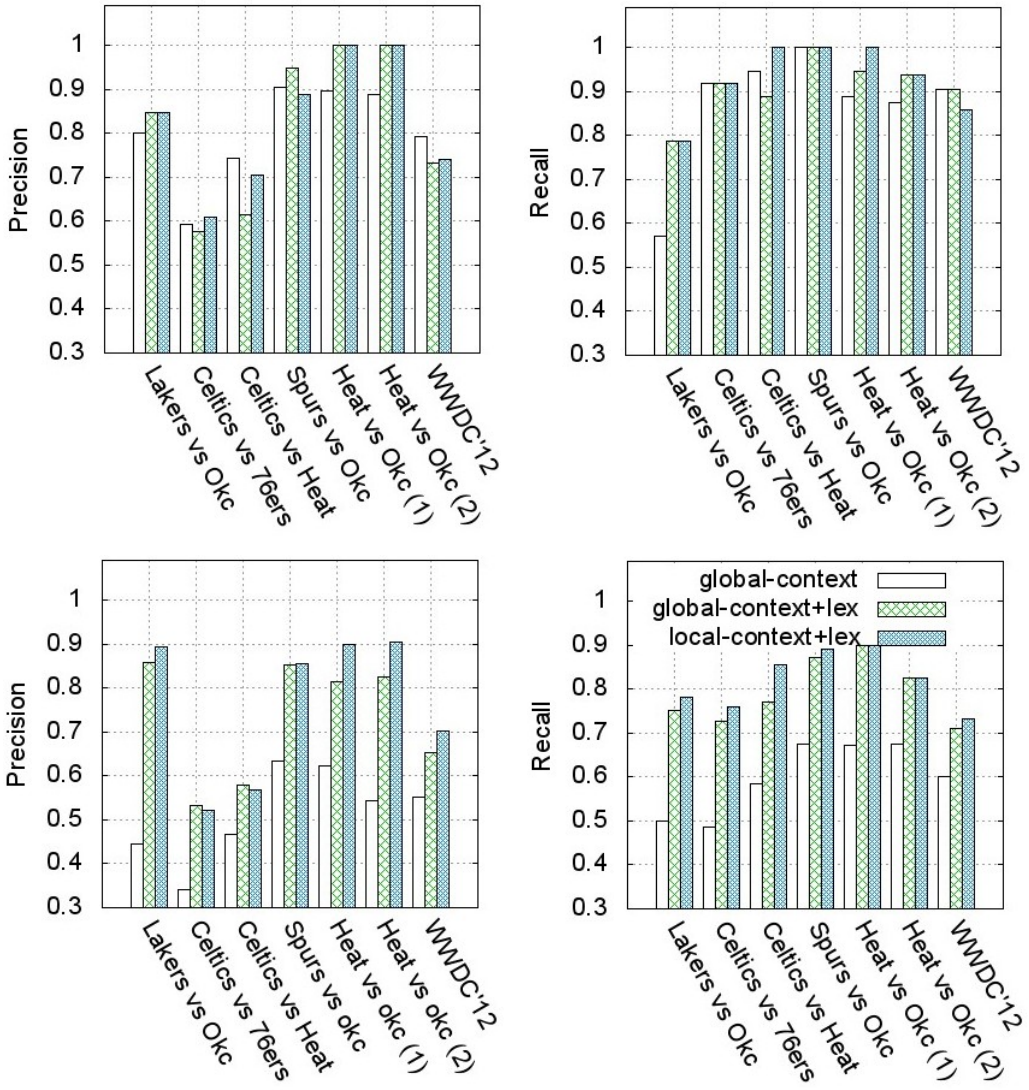


Figure 4.3: Participant detection performance. The upper figures represent the participant-level precision and recall scores, while the lower figures represent the mention-level precision and recall. X-axis corresponds to the six NBA games and the WWDC conference.

below:

$$\text{participant-prec} = \#\text{correct-participants}/|T_s|$$

$$\text{participant-recall} = \#\text{hit-participants}/|T_g|$$

Note that a correct participant may include incorrect mentions, and that more than one correct participants may correspond to the same hit participant, both of which are undesired. In the latter case, we use **representative participant** to refer to the correct participant which contains the most mentions in the hit participant. In this way, we build a 1-to-1 mapping from the detected participants to the groundtruth participants. Next, we define **correct mentions** as the union of the overlapping mentions between all pairs of representative and hit participants. Then we calculate the mention-level precision and recall as the number of correct mentions divided by the total mentions in the system or goldstandard participant clusters.

Figure 4.3 shows the participant- and mention-level precision and recall scores. We experimented with different similarity measures for the agglomerative clustering approach[10]. The "global context" means that the context vectors are created from the entire data stream; this may not perform well since different participants can share similar global context. E.g., the terms "shot", "dunk", "rebound" can appear in the context of any NBA players and are not discriminative enough. We found that adding the lexical similarity measure greatly boosted the clustering performance, especially on the mention-level, and that combining the lexical similarity with the local context is even more helpful for some events. We notice that two event topics (celtics vs 76ers and celtics vs heat) yield relatively low precision on both participant- and mention-level. Taking a close look at the data, we found that these two event topics accidentally co-occurred with other popular event topics, namely the TV program "American Idol" finale and the NBA Draft. The

[10]The stopping threshold δ was set to 0.15, local context length is 3 minutes, and frequency threshold ψ was set to 200.

keyword based data crawler thus includes many noisy tweets in the event streams, leading to some false participants being detected.

4.3.3 Event Detection Results

Event	Participant-level Event Detection							
	#P	#S	Spike			MM		
			R	P	F	R	P	F
Lakers vs Okc	9	65	0.75	0.31	**0.44**	0.71	0.39	**0.50**
Celtics vs 76ers	10	88	0.52	0.39	**0.45**	0.53	0.43	**0.47**
Celtics vs Heat	14	152	0.53	0.29	**0.37**	0.50	0.38	**0.43**
Spurs vs Okc	12	98	0.78	0.46	**0.58**	0.84	0.57	**0.68**
Heat vs Okc (1)	15	123	0.75	0.27	**0.40**	0.72	0.35	**0.47**
Heat vs okc (2)	13	153	0.74	0.36	**0.48**	0.76	0.43	**0.55**
WWDC'12	10	56	0.64	0.14	**0.23**	0.59	0.33	**0.42**
Average	**12**	**105**	0.67	0.32	**0.42**	0.66	0.41	**0.50**

Table 4.4: Event detection results on participant streams.

Event	Global Event Detection									
	#S	Spike			Participant + Spike			Participant + MM		
		R	P	F	R	P	F	R	P	F
Lakers vs Okc	48	0.67	0.38	0.48	**0.94**	0.19	0.32	**0.88**	0.40	**0.55**
Celtics vs 76ers	60	0.65	0.51	0.57	**0.72**	0.18	0.29	**0.78**	0.39	**0.52**
Celtics vs Heat	67	0.57	0.41	0.48	**0.97**	0.21	0.35	**0.91**	0.28	**0.43**
Spurs vs Okc	81	0.41	0.42	0.41	**0.88**	0.35	0.50	**0.91**	0.54	**0.68**
Heat vs Okc (1)	85	0.41	0.47	0.44	**0.94**	0.20	0.33	**0.96**	0.34	**0.50**
Heat vs okc (2)	92	0.41	0.33	0.37	**0.88**	0.21	0.34	**0.87**	0.38	**0.53**
WWDC'12	43	0.53	0.26	0.35	**0.77**	0.14	0.24	**0.70**	0.31	**0.43**
Average	**68**	0.52	0.40	0.44	**0.87**	0.21	0.34	**0.86**	0.38	**0.52**

Table 4.5: Event detection results on the input streams.

We compare our proposed time-content mixture model (noted as "MM") against the spike detection algorithm proposed in [MBB+11] (noted as "Spike") . The spike algorithm is based on the tweet volume change. It uses 10 seconds as a time unit, calculates the tweet arrival rate in each unit, and identifies the rates that are significantly higher than

the mean tweet rate. For these rate spikes, the algorithm finds the local maximum of tweet rate and identify a window surrounding the local maximum. We tune the parameter of the "Spike" approach (set $\tau = 4$) so that it yields similar recall values as the mixture model approach. We then apply the "MM" and "Spike" approaches to both the participant and event streams and evaluate the event detection performance. Results are shown in Table 4.4. A system detected event is considered to match the goldstandard event if its peak time is within a 2-minute window of the goldstandard.

We first apply the "Spike" and "MM" approach to the participant streams. The participant streams on which we cannot detect any meaningful events have been excluded, the resulting number of participants are listed in Table 4.4 and denoted as "#P", and "#S" is the summation number of events from all participant streams of each input dataset. In general, we found the "MM" approach can perform better since it inherently incorporates both the "burstiness" and "lexical cohesiveness" of the event tweets, while the "Spike" approach relies solely on the "burstiness" property. Note that although we divide the entire event stream into participant streams, some key participants still own huge amount of discussion and the spike patterns are not always clearly identifiable. The time-content mixture model gains advantages in these cases.

We apply three settings to detect global events on the data streams in Table 4.5. "Spike" directly applies the spike algorithm on the entire event stream; the "Participant + Spike" and "Participant + MM" approaches first perform event detection on the participant streams and then merge the detected events along the timeline to generate global events. Note that there are fewer goldstandard events ("#S") on the global streams since each global event may correspond to one or multiple participant-level events. Because of the averaging effect, spike patterns on the entire event stream is less obvious than those on the participant streams. As a result, few spikes have been detected on the event stream using the "Spike" algorithm, which leads to low recall as compared to other participant-

based approaches. It also indicates that, by dividing the entire event stream into participant streams, we have a better chance of identifying the events that have otherwise been shadowed by the dominant events or participants. The two participant-based methods yield similar recall but "Participant + Spike" yields slightly worse precision, since it is very sensitive to the spikes on the participant-level, leading to the rise of false alarms. The "Participant + MM" approach is much better in precision, which is consistent to our findings on the participant streams.

4.4 Summary

Event detection is critical for text analysis of social media streams to capture the event-related information. Existing methods reply on the volume change of the whole stream to detect bursts or spikes. In this chapter, we propose a method which first divides the whole stream into several participants streams, and then combines the information of volume changes of the stream and topic changes. Experiments demonstrate that the proposed method leads to more robust detection results.

CHAPTER 5

MULTI-DOCUMENT SUMMARIZATION

5.1 Multi-document Summarization using Dominating Set

5.1.1 Introduction

Multi-document summarization is a useful tool to address the information overload problem, which can be classified into extractive and abstractive summarization[Man01]. Extractive summarization methods select important sentences from the original documents, while abstractive summarization methods attempt to rephrase the information in the text. For different information needs, different summaries should be generated as different views of the data set. In this dissertation, we focus on four types of summarization.

In this dissertation, we propose a new principled and versatile framework for multi-document summarization using the *minimum dominating set*. Many known summarization tasks including generic, query-focused, update, and comparative summarization can be modeled as different variations derived from the proposed framework. The framework provides an elegant basis to establish the connections between various summarization tasks while highlighting their differences.

In our framework, a sentence graph is first generated from the input documents where vertices represent sentences and edges indicate that the corresponding vertices are similar. A natural method for describing the extracted summary is based on the idea of graph domination [WL01]. *A dominating set* of a graph is a subset of vertices such that every vertex in the graph is either in the subset or adjacent to a vertex in the subset; and *a minimum dominating set* is a dominating set with the minimum size. The minimum dominating set of the sentence graph can be naturally used to describe the summary: it is *representative* since each sentence is either in the minimum dominating set or connected to one sentence

in the set; and it is with *minimal redundancy* since the set is of minimum size. Approximation algorithms are proposed for performing summarization and empirical experiments are conducted to demonstrate the effectiveness of our proposed framework. Though the dominating set problem has been widely used in wireless networks, this paper is the first work on using it for modeling sentence extraction in document summarization.

5.1.2 Related Work

Query-Focused Summarization In query-focused summarization, the information of the given topic or query should be incorporated into summarizers, and sentences suiting the user's declared information need should be extracted. Many methods for generic summarization can be extended to incorporate the query information [SBC03, WLLH08]. [WYX07a] made full use of both the relationships among all the sentences in the documents and relationship between the given query and the sentences by manifold ranking. Probability models have also been proposed with different assumptions on the generation process of the documents and the queries [DIM06, HV09, TYC09].

Update Summarization and Comparative Summarization Update summarization was introduced in Document Understanding Conference (DUC) 2007 [Dan07] and was a main task of the summarization track in Text Analysis Conference (TAC) 2008 [DO08]. It is required to summarize a set of documents under the assumption that the reader has already read and summarized the first set of documents as the main summary. To produce the update summary, some strategies are required to avoid redundant information which has already been covered by the main summary. One of the most frequently used methods for removing redundancy is Maximal Marginal Relevance(MMR) [GMCK00]. Comparative document summarization was proposed in [WZLG09a] to summarize the differences between comparable document groups. A sentence selection approach was proposed in

[WZLG09a] to accurately discriminate the documents in different groups modeled by the conditional entropy.

Dominating Set Many approximation algorithms have been developed for finding minimum dominating set for a given graph [GK98, TZTX07]. Kann [Kan92] show that the minimum dominating set problem is equivalent to set cover problem, which is a well-known NP-hard problem. Dominating set has been widely used for clustering in wireless networks [CL02, HJ07]. It has been used to find topic words for hierarchical summarization [LCR01], where a set of topic words is extracted as a dominating set of word graph. In our work, we use the minimum dominating set to formalize the sentence extraction for document summarization.

5.1.3 The Summarization Framework

Sentence Graph Generation

To perform multi-document summarization via minimum dominating set, we need to first construct a sentence graph in which each node is a sentence in the document collection. In our work, we represent the sentences as vectors based on tf-isf, and then obtain the cosine similarity for each pair of sentences. If the similarity between a pair of sentences s_i and s_j is above a given threshold λ, then there is an edge between s_i and s_j.

For generic summarization, we use all sentences for building the sentence graph. For query-focused summarization, we only use the sentences containing at least one term in the query. In addition, when a query q is involved, we assign each node s_i a weight, $w(s_i) = d(s_i, q) = 1 - \cos(s_i, q)$, to indicate the distance between the sentence and the query q.

After building the sentence graph, we can formulate the summarization problem using the minimum dominating set. A graphical illustration of the proposed framework is shown in Figure 5.1.

The Minimum Dominating Set Problem

Given a graph $G =< V, E >$, *a dominating set* of G is a subset S of vertices with the following property: each vertex of G is either in the dominating set S, or is adjacent to some vertices in S.

Problem 5.1.1 *Given a graph G, the minimum dominating set problem (MDS) is to find a minimum size subset S of vertices, such that S forms a dominating set.*

MDS is closely related to the set cover problem (SC), a well-known NP-hard problem.

Problem 5.1.2 *Given F, a finite collection $\{S_1, S_2, \ldots, S_n\}$ of finite sets, the set cover problem (SC) is to find the optimal solution*

$$F^* = \arg\min_{F' \subseteq F} |F'| \text{ s.t. } \bigcup_{S' \in F'} S' = \bigcup_{S \in F} S.$$

Theorem 5.1.3 *There exists a pair of polynomial time reduction between MDS and SC.*

Proof. Here we sketch the proof. To reduce from the minimum dominating set problem to SC, For each input of the minimum dominating set problem, a graph $G =< V, E >$ with $V = \{1, \ldots, n\}$, we can construct a finite collection of finite sets $F = \{S_1, S_2, \ldots, S_n\}$ by defining $S_i = \{i\} \cup \{j \in [1..n] : (i, j) \in E\}$. A vertex $i \in V$ can be covered either by including S_i, corresponding to including the node i in the dominating set, or by including one of the sets S_j such that $(i, j) \in E$, corresponding to including node j in the dominating set. Thus the minimum dominating set $D^* \subseteq V$ gives us the minimum set cover F^* of the same size and every set cover of F gives us a dominating set of G. So

we have obtained a polynomial L-reduction from the minimum dominating set problem to SC. Similarly, we can show that there is a polynomial time L-reduction from SC to the minimum dominating set problems. More details can been found in [Kan92]. □

So, MDS is also NP-hard and it has been shown that there are no approximate solutions within $c \log |V|$, for some $c > 0$ [Fei98, RS97].

An Approximation Algorithm A greedy approximation algorithm for the SC problem is described in [Joh73]. Basically, at each stage, the greedy algorithm chooses the set which contains the largest number of uncovered elements.

Based on Theorem 5.1.3, we can obtain a greedy approximation algorithm for MDS. Starting from an empty set, if the current subset of vertices is not the dominating set, a new vertex which has the most number of the adjacent vertices that are not adjacent to any vertex in the current set will be added.

Proposition 5.1.4 *The greedy algorithm approximates SC within* $1 + \ln s$ *where* s *is the size of the largest set.*

It was shown in [Joh73] that the approximation factor for the greedy algorithm is no more than $H(s)$, the s-th harmonic number:

$$H(s) = \sum_{k=1}^{s} \frac{1}{k} \leq \ln s + 1$$

Corollary 5.1.5 *MDS has a approximation algorithm within* $1 + \ln \Delta$ *where* Δ *is the maximum degree of the graph.*

Corollary 5.1.5 follows directly from Theorem 5.1.3 and Proposition 5.1.4.

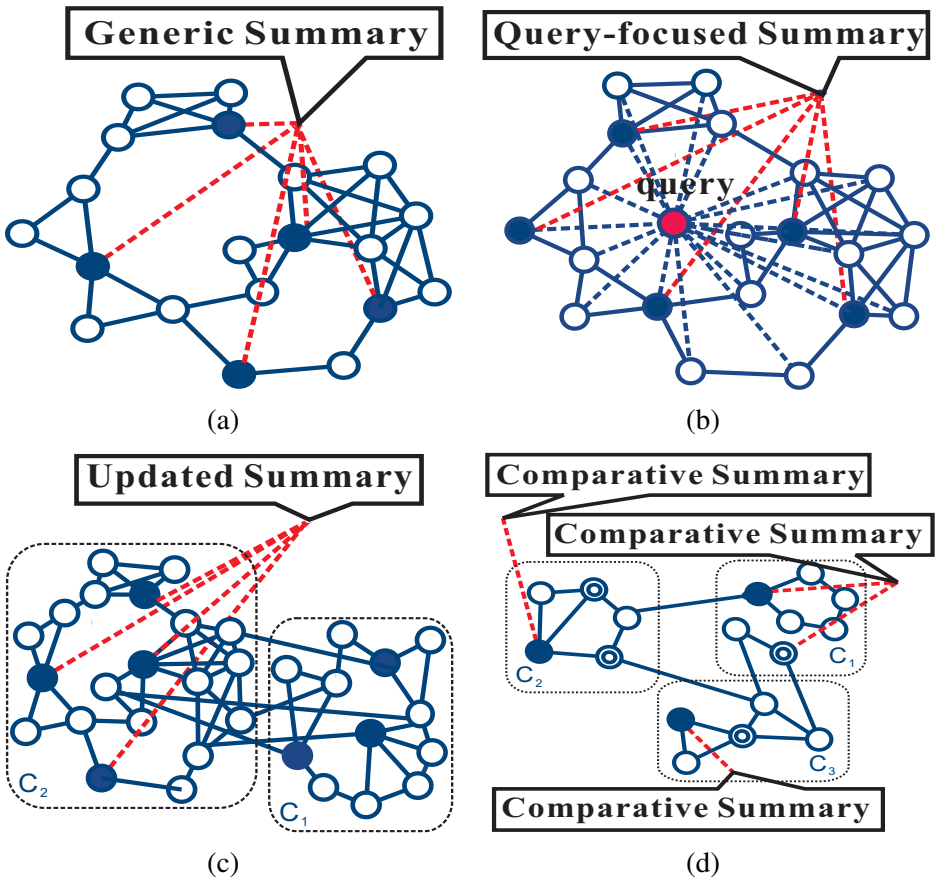


Figure 5.1: Graphical illustrations of multi-document summarization via the minimum dominating set.

Generic Summarization

Generic summarization is to extract the most representative sentences to capture the important content of the input documents. Without taking into account the length limitation of the summary, we can assume that the summary should represent all the sentences in the document set (i.e., every sentence in the document set should either be extracted or be similar with one extracted sentence). Meanwhile, a summary should also be as short as possible. Such summary of the input documents under the assumption is exactly the minimum dominating set of the sentence graph we constructed from the input documents in Section 5.1.3. Therefore the summarization problem can be formulated as the minimum dominating set problem.

Algorithm 2 Algorithm for Generic Summarization

INPUT: G, W
OUTPUT: S

1: $S = \emptyset$
2: $T = \emptyset$
3: **while** $L(S) < W$ and $V(G)! = S$ **do**
4: **for** $v \in V(G) - S$ **do**
5: $s(v) = |\{ADJ(v) - T\}|$
6: $v^* = \arg\max_v s(v)$
7: $S = S \cup \{v^*\}$
8: $T = T \cup ADJ(v^*)$

However, usually there is a length restriction for generating the summary. Moreover, the MDS is NP-hard as shown in Section 5.1.3. Therefore, it is straightforward to use a greedy approximation algorithm to construct a subset of the dominating set as the final summary. In the greedy approach, at each stage, a sentence which is optimal according to the local criteria will be extracted. Algorithm 2 describes an approximation algorithm for generic summarization. In Algorithm 2, G is the sentence graph, $L(S)$ is the length of the summary, W is the maximal length of the summary, and $ADJ(v) = \{v' | (v', v) \in E(G)\}$

is the set of vertices which are adjacent to the vertex v. A graphical illustration of generic summarization using the minimum dominating set is shown in Figure 5.1(a).

Query-Focused Summarization

Letting G be the sentence graph constructed in Section 5.1.3 and q be the query, the query-focused summarization can be modeled as

$$D^* = \arg\min_{D \subseteq G} \sum_{s \in D} d(s, q) \quad \text{s.t. } D \text{ is a dominating set of } G. \tag{5.1}$$

Note that $d(s, q)$ can be viewed as the weight of vertex in G. Here the summary length is minimized implicitly, since if $D' \subseteq D$, then $\sum_{s \in D'} d(s, q) \leq \sum_{s \in D} d(s, q)$. The problem in Eq.(5.1) is exactly a variant of the minimum dominating set problem, i.e., the minimum weighted dominating set problem (MWDS).

Similar to MDS, MWDS can be reduced from the weighted version of the SC problem. In the weighted version of SC, each set has a weight and the sum of weights of selected sets needs to be minimized. To generate an approximate solution for the weighted SC problem, instead of choosing a set i maximizing $|SET(i)|$, a set i minimizing $\frac{w(i)}{|SET(i)|}$ is chosen, where $SET(i)$ is composed of uncovered elements in set i, and $w(i)$ is the weight of set i. The approximate solution has the same approximation ratio as that for MDS, as stated by the following theorem [Chv79].

Theorem 5.1.6 *An approximate weighted dominating set can be generated with a size at most* $1 + \log \Delta \cdot |OPT|$*, where* Δ *is the maximal degree of the graph and* OPT *is the optimal weighted dominating set.*

Accordingly, from generic summarization to query-focused summarization, we just need to modify line 6 in Algorithm 2 to

$$v^* = \arg\min_v \frac{w(v)}{s(v)}, \tag{5.2}$$

where $w(v)$ is the weight of vertex v. A graphical illustration of query-focused summarization using the minimum dominating set is shown in Figure 5.1(b).

Update Summarization

Give a query q and two sets of documents C_1 and C_2, update summarization is to generate a summary of C_2 based on q, given C_1. Firstly, summary of C_1, referred as D_1 can be generated. Then, to generate the update summary of C_2, referred as D_2, we assume D_1 and D_2 should represent all query related sentences in C_2, and length of D_2 should be minimized.

Let G_1 be the sentence graph for C_1. First we use the method described in Section 5.1.3 to extract sentences from G_1 to form D_1. Then we expand G_1 to the whole graph G using the second set of documents C_2. G is then the graph presentation of the document set including C_1 and C_2. We can model the update summary of C_2 as

$$D^* = \arg\min_{D_2} \sum_{s \in D_2} w(s) \tag{5.3}$$
$$\text{s.t. } D_2 \cup D_1 \text{ is a dominating set of } G.$$

Intuitively, we extract the smallest set of sentences that are closely related to the query from C_2 to complete the partial dominating set of G generated from D_1. A graphical illustration of update summarization using the minimum dominating set is shown in Figure 5.1(c), where vertices in the right rectangle represent the first document set C_1, and ones in the left represent the second document set where update summary is generated..

Comparative Summarization

Comparative document summarization aims to summarize the differences among comparable document groups. The summary produced for each group should emphasize its difference from other groups [WZLG09a].

We extend our method for update summarization to generate the discriminant summary for each group of documents. Given N groups of documents $C_1, C_2, \ldots, C_N$, we first generate the sentence graphs $G_1, G_2, \ldots, G_N$, respectively. To generate the summary for $C_i, 1 \leq i \leq N$, we view C_i as the update of all other groups. To extract a new sentence, only the one connected with the largest number of sentences which have no representatives in any groups will be extracted. We denote the extracted set as the complementary dominating set, since for each group we obtain a subset of vertices dominating those are not dominated by the dominating sets of other groups. To perform comparative summarization, we first extract the standard dominating sets for $G_1, \ldots, G_N$, respectively, denoted as D_1, ..., D_N. Then we extract the so-called complementary dominating set CD_i for G_i by continuing adding vertices in G_i to find the dominating set of $\cup_{1 \leq j \leq N} G_j$ given $D_1, \ldots, D_{i-1}, D_{i+1}, \ldots, D_N$. A graphical illustration of comparative summarization is shown in Figure 5.1(d), where each rectangle represents a group of documents, and vertices with rings are the dominating set for each group, while the solid vertices are the complementary dominating set, which is extracted as comparative summaries.

5.1.4 Experiments

Data Sets

In the experiments, we evaluate the proposed framework on news data from DUC/TAC which is widely used as benchmarks in the summarization community for the generic,

Data set	Type of Summarization	#Topics	#Documents/topic	Summary length
DUC04	Generic	40	10	665 bytes
DUC05	Topic-focused	50	25	250 words
DUC06	Topic-focused	50	25	250 words
TAC08 A	Topic-focused	48	10	100 words
TAC08 B	Update	48	10	100 words

Table 5.1: Brief description of the data set

query-focused and update summarization tasks, and blog data for comparative summarization.

Table 5.1 shows the characteristics of the data sets. We use DUC04 data set to evaluate our method for generic summarization task and DUC05 and DUC06 data sets for query-focused summarization task. The data set for update summarization, (i.e. the main task of TAC 2008 summarization track) consists of 48 topics and 20 newswire articles for each topic. The 20 articles are grouped into two clusters. The task requires to produce 2 summaries, including the initial summary (TAC08 A) which is standard query-focused summarization and the update summary (TAC08 B) under the assumption that the reader has already read the first 10 documents.

Evaluation Metrics

We use ROUGE [LH03] toolkit (version 1.5.5) to measure the summarization performance, which is widely applied by DUC for performance evaluation. It measures the quality of a summary by counting the unit overlaps between the candidate summary and a set of reference summaries. Several automatic evaluation methods are implemented in ROUGE, such as ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-SU. ROUGE-N is an n-gram recall computed as follows.

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{ref}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{ref}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)} \tag{5.4}$$

	DUC04	
	ROUGE-2	ROUGE-SU
DUC Best	0.09216	0.13233
Centroid	0.07379	0.12511
LexPageRank	0.08572	0.13097
BSTM	0.09010	0.13218
MDS	0.08934	0.13137

Table 5.2: Results on generic summarization.

	DUC05		DUC06	
	ROUGE-2	ROUGE-SU	ROUGE-2	ROUGE-SU
DUC Best	0.0725	0.1316	0.09510	0.15470
SNMF	0.06043	0.12298	0.08549	0.13981
TMR	0.07147	0.13038	0.09132	0.15037
Wiki	0.07074	0.13002	0.08091	0.14022
MWDS	0.07311	0.13061	0.09296	0.14797

Table 5.3: Results on query-focused summarization.

where n is the length of the n-gram, and ref stands for the reference summaries. $\text{Count}_{\text{match}}(\text{gram}_n)$ is the maximum number of n-grams co-occurring in a candidate summary and the reference summaries, and $\text{Count}(\text{gram}_n)$ is the number of n-grams in the reference summaries. ROUGE-L uses the longest common subsequence (LCS) statistics, while ROUGE-W is based on weighted LCS and ROUGE-SU is based on skip-bigram plus unigram. Each of these evaluation methods in ROUGE can generate three scores (recall, precision and F-measure). As we have similar conclusions in terms of any of the three scores, for simplicity, in this paper, we only report the average F-measure scores generated by ROUGE-1, ROUGE-2,ROUGE-L,ROUGE-W and ROUGE-SU4 (where skip length is 4) to compare the implemented systems.

We apply a 5-fold cross-validation procedure to choose the threshold λ used for generating the sentence graph in our method.

Generic Summarization

We implement the following widely used or recent published methods for generic summarization as the baseline systems to compare with our proposed method (denoted as MDS). (1) Centroid: The method applies MEAD algorithm [RJST04] to extract sentences according to the following three parameters: centroid value, positional value, and first-sentence overlap. (2) LexPageRank: The method first constructs a sentence connectivity graph based on cosine similarity and then selects important sentences based on the concept of eigenvector centrality [ER04]. (3) BSTM: A Bayesian sentence-based topic model making use of both the term-document and term-sentence associations [WZLG09b].

Our method outperforms the simple Centroid method and another graph-based LexPageRank, and its performance is close to the results of the Bayesian sentence-based topic model and those of the best team in the DUC competition. Note however that, like clustering or topic based methods, BSTM needs the topic number as the input, which usually varies by different summarization tasks and is hard to estimate.

Query-Focused Summarization

We compare our method (denoted as MWDS) described in Section 5.1.3 with some recently published systems. (1) TMR [TYC09]: incorporates the query information into the topic model, and uses topic based score and term frequency to estimate the importance of the sentences. (2) SNMF [WLZD08]: calculates sentence-sentence similarities by sentence-level semantic analysis, clusters the sentences via symmetric nonnegative matrix factorization, and extracts the sentences based on the clustering result. (3) Wiki [Nas08]: uses Wikipedia as external knowledge to expand query and builds the connection between the query and the sentences in documents.

Table 5.3 presents the experimental comparison of query-focused summarization on the two datasets. From Table 5.3, we observe that our method is comparable with these

systems. This is due to the good interpretation of the summary extracted by our method, an approximate minimal dominating set of the sentence graph. On DUC05, our method achieves the best result; and on DUC06, our method outperforms all other systems except the best team in DUC. Note that our method based on the minimum dominating set is much simpler than other systems. Our method only depends on the distance to the query and has only one parameter (i.e., the threshold λ in generating the sentence graph).

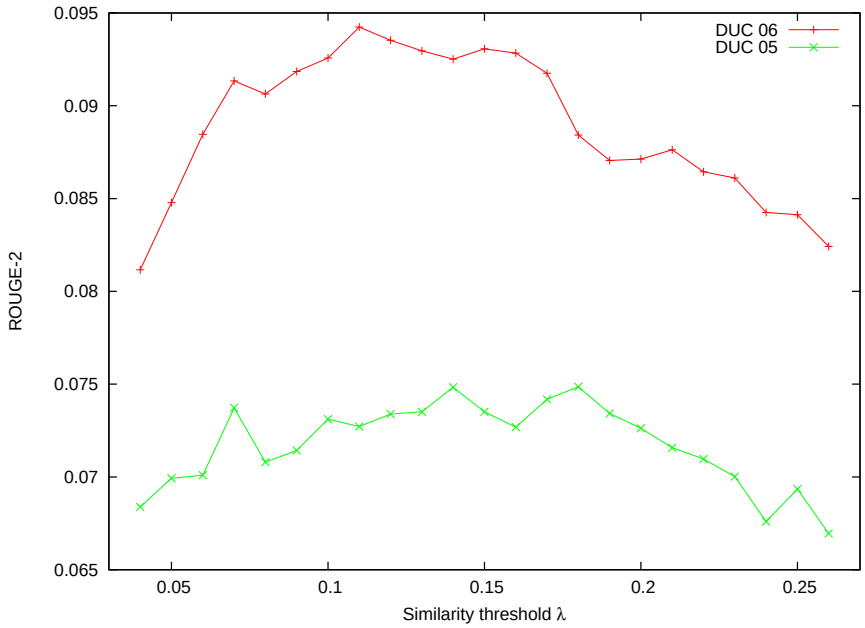


Figure 5.2: ROUGE-2 vs. threshold λ

We also conduct experiments to empirically evaluate the sensitivity of the threshold λ. Figure 5.2 shows the ROUGE-2 curve of our MWDS method on the two datasets when λ varies from 0.04 to 0.26. When λ is small, edges fail to represent the similarity of the sentences, while if λ is too large, the graph will be sparse. As λ is approximately in the range of $0.1 - 0.17$, ROUGE-2 value becomes stable and relatively high.

Update Summarization

Table 5.4 presents the experimental results on update summarization. In Table 5.4, 'TAC Best" and "TAC Median" represent the best and median results from the participants of TAC 2008 summarization track in the two tasks respectively according to the TAC 2008 report [DO08]. As seen from the results, the ROUGE scores of our methods are higher than the median results. The good results of the best team typically come from the fact that they utilize advanced natural language processing (NLP) techniques to resolve pronouns and other anaphoric expressions. Although we can spend more efforts on the preprocessing or language processing step, our goal here is to demonstrate the effectiveness of formalizing the update summarization problem using the minimum dominating set and hence we do not utilize advanced NLP techniques for preprocessing. The experimental results demonstrate that our simple update summarization method based on the minimum dominating set can lead to competitive performance for update summarization.

	TAC08 A		TAC08 B	
	ROUGE-2	ROUGE-SU	ROUGE-2	ROUGE-SU
TAC Best	0.1114	0.14298	0.10108	0.13669
TAC Median	0.08123	0.11975	0.06927	0.11046
MWDS	0.09012	0.12094	0.08117	0.11728

Table 5.4: Results on update summarization.

Comparative Summarization

We use the top six largest clusters of documents from TDT2 corpora to compare the summary generated by different comparative summarization methods. The topics of the six document clusters are as follows: topic 1: Iraq Issues; topic 2: Asia's economic crisis; topic 3: Lewinsky scandal; topic 4: Nagano Olympic Games; topic 5: Nuclear Issues in Indian and Pakistan; and topic 6: Jakarta Riot. From each of the topics, 30 documents are extracted randomly to produce a one-sentence summary. For comparison purpose, we

Topic	Complementary Dominating Set	Discriminative Sentence Selection	Dominating Set
1	$\cdots$ U.S. Secretary of State Madeleine Albright arrives to consult on the stand-off between the **United Nations and Iraq**.	**the U.S. envoy to the United Nations**, Bill Richardson, $\cdots$ play down China's refusal to support threats of **military force against Iraq**	The United States and Britain do not trust **President Saddam** and wants $cdots$warning of serious consequences if **Iraq** violates the accord.
2	**Thailand's currency**, the baht, **dropped through a key psychological level** of $\cdots$ amid a regional sell-off sparked by escalating **social unrest in Indonesia**.	Earlier, driven largely by **the declining yen**, **South Korea's stock market fell** by $\cdots$, while the **Nikkei 225 benchmark index dipped** below 15,000 in the morning $\cdots$	*In the fourth quarter, IBM Corp. earned $2.1 billion, up 3.4 percent from $2 billion a year earlier.*
3	$\cdots$ attorneys representing **President Clinton and Monica Lewinsky.**	The following night **Isikoff** $\cdots$, where he directly followed the recitation of the top-10 list: "Top 10 **White House Jobs That Sound Dirty.**"	In Washington, Ken Starr's grand jury continued its investigation of the **Monica Lewinsky matter**.
4	Eight women and six men were named Saturday night as the first **U.S. Olympic Snowboard Team** as their sport gets set to make its debut in **Nagano, Japan**.	*this tunnel is finland's cross country version of tokyo's alpine ski dome, and olympic skiers flock from russia, $\cdots$, france and austria this past summer to work out the kinks* $\cdots$	If the skiers the **men's super-G** and the **women's downhill** on Saturday, they will be back on schedule.
5	**U.S. officials** have announced **sanctions** Washington will impose on **India and Pakistan** for conducting **nuclear tests**.	The **sanctions** would stop all foreign aid except for humanitarian purposes, **ban military sales to India** $\cdots$	And **Pakistan's prime minister** says his country will sign the **U.N.'s comprehensive ban on nuclear tests** if **India** does, too.
6	$\cdots$ remain in force around **Jakarta**, and at the Parliament building where **thousands of students staged a sit-in** Tuesday $\cdots$.	"**President Suharto** has given much to his country over the past 30 years, raising **Indonesia's** standing in the world $\cdots$	*What were the students doing at the time you were there, and what was the reaction of the students to the troops?*

Table 5.5: A case study on comparative document summarization.

extract the sentence with the maximal degree as the baseline. Note that the baseline can be thought as an approximation of the dominating set using only one sentence. Table 5.5 shows the summaries generated by our method (complementary dominating set (CDS)), discriminative sentence selection (DSS) [WZLG09a] and the baseline method. Some unimportant words are skipped due to the space limit. The bold font is used to annotate the phrases that are highly related with the topics, and italic font is used to highlight the sentences that are not proper to be used in the summary. Our CDS method can extract discriminative sentences for all the topics. DSS can extract discriminative sentences for all the topics except topic 4. Note that the sentence extracted by DSS for topic 4 may be discriminative from other topics, but it is deviated from the topic Nagano Olympic Games. In addition, DSS tends to select long sentences which should not be preferred for summarization purpose. The baseline method may extract some general sentences, such as the sentence for topic 2 and topic 6 in Table 5.5.

5.2 Multi-document Summarization Using Learning-to-Rank

As a fundamental and effective tool for document understanding, organization, and navigation, query-focused multi-document summarization has been very active and enjoying a growing amount of attention with the ever-increasing growth of the social media document data (e.g., blogs, tweets). For query-focused multi-document summarization, a summarizer incorporates user declared queries and generates summaries that not only reflect the important concepts in the input documents but also bias to the queries. Query-focused multi-document summarization methods can be broadly classified into two types: extractive summarization and abstractive summarization. Extractive summarization usually selects phrases or sentences from the input documents while abstractive summarization involves paraphrasing components of input documents and sentence reformulation [KM02].

There are many recent studies on query-focused multi-document summarization and most proposed techniques are extractive methods. Typical examples include methods based on knowledge in Wikipedia [Nas08], information distance [LHZL09], non-negative matrix factorization [WLZD08], graph theory [SL10] and graph ranking [OER05, WYX07a].

Generally speaking, the extracted sentences in the summary should be *representative* or *salient*, capturing the important content related to the queries with *minimal redundancy* [JM08]. In particular, these extractive summarization methods typically select the sentences in the input documents to form the summary based on a set of content or linguistic features, such as term frequency-inverse sentence frequency (tf-isf), sentence or term position, salient or informative keywords, and discourse information. Various features have been used to characterize the different aspects of the sentences and their relevance to the queries.

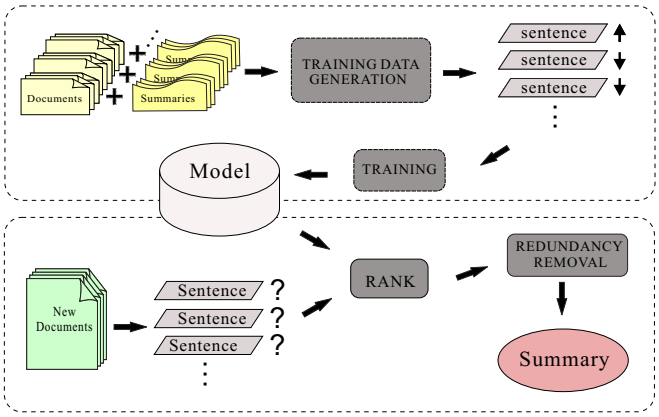


Figure 5.3: The framework of supervised learning for summarization.

Supervised Learning for Summarization

By composing manual summaries, we can naturally create labeling data of query-focused multi-document summarization is in the form of triples <query, document set, human summaries>. However, in order to make use of this kind of data, and apply a standard supervised learning algorithm (classification/regression/ranking) to learn a model to rank the sentences for a new <query, document set> pair, the existing human labeling data needs to be transformed first to generate the training data for supervised learning, that is, to assign a label/score for each sentence. The general framework of an extractive summarization system using supervised learning is given in Figure 5.3. The framework consists of the following major components: (1) training data generation where the given human summaries are transformed into the training data for supervised learning; (2) model learning where a supervised learning model is constructed to label/rank the sentences; and (3) summary generation for new documents where the learned model is used for ranking the sentences followed by redundancy removal. Note the data transformation is not trivial, because human-generated summaries are abstractive and do not necessarily well match the sentences in the documents. To solve this problem, in this paper, both the training data generation and the subsequent model learning component are considered.

Recently, support vector regression (SVR), has been used to automatically combine various sentence features for supervised summarization [OLL07]. However, since we only need to differentiate the "summary sentence" and "non-summary sentence", the model is not necessary to fit the regression scores of the training data. In other words, it should make no difference if we swap two non-summary sentences which are ranked low in a ranked sentence list, even thougth their regression scores are different. So the objective in regression model learning is too aggressive, measuring the average distance between the predicted score and the true score for all sentences. Another reason of the problem of regression model is that the true score for a sentence in the training set is estimated automatically and the quality of the estimation is not guaranteed.

In this chapter, we proposes a method for text summarization based on ranking techniques and explore the use of ranking SVM [Joa02], a learning to rank method, to train the feature weights for query-focused multi-document summarization. To construct the training data for ranking SVM, a rank label of "summary sentence" or "non-summary sentence" needs to be assigned to the training sentences. This assignment generally relies on a threshold of sentence scoring. Our experiments show that a small variation of the threshold may lead to a substantial change on the performance of the trained model. The sentences near the threshold are likely to be assigned with a wrong rank label, thus, introducing noise into the training set. To make the threshold less sensitive, we adopt a cost sensitive loss in the ranking SVM's objective function, giving less weights to those sentence pairs whose relative positions are of less certainty. While there are existing works on using ranking for summarization, the proposed method of cost sensitive loss will improve the robustness of learning and extend the usefulness of rank-based summarization techniques.

Our work also contribute to training data generation for supervised summarization. Note that the problem of automatic training data generation is essential in trainable sum-

marizers. To better estimate the probability of a sentence in the document set to be a summary sentence, we proposes a novel method by utilizing the sentence relationships to improve the estimation of the probability in training data generation.

5.2.1 Related Work

Supervised Learning for Summarization

Supervised learning approaches have been successfully applied in single document summarization, where the training data is available or easy to build. The most straightforward way is to regard the sentence extraction task as a binary classification problem. [KPC95] developed a trainable summarization system which adopted various features and used a Bayesian classifier to learn the feature weights. The system performed better than other systems using only a single feature. [HIMM02] trained a SVM model for important sentence extraction and the model outperformed other classification models such as decision-tree or boosting methods on the Japanese Text Summarization Challenge (TSC). To make use of the sentence relations in a single document, sequential labeling methods are used to extract a summary for a single document. [ZH03] applied a HMM-based model and [SSL+07] proposed a conditional random field based framework.

For query-focused multi-document summarization, [ZHW05] applied the Conditional Maximum Entropy, a classification model, on the DUC 2005 query-based summarization task. Similar to those methods developed for single document summarization, the model was trained on an existing training dataset where sentences are labeled as summary or non-summary manually. [OLL07] constructed the training data by labeling the sentence with a "true" score calculated according to human summaries, and then used support vector regression (SVR) to relate the "true" score of the sentence to its features. Similar to [OLL07], in this paper, we construct the training data from human summaries. How-

ever, the learning to rank method is used in our work for query-focused multi-document summarization.

Learning to Rank

Learning to rank, in parallel with learning for classification and regression, has been attracting increasing interests in statistical learning for the last decade, because many applications such as web search and retrieval can be formalized as ranking problems.

Many of the learning to rank approaches are pairwise approaches, where the learning to rank problem is approximated by a classification problem, and a classifier is learned to tell whether a document is better than another. Recently, a number of authors have proposed directly defining a loss function on a list of objects and directly optimizing the loss function in learning [CQL$^+$07, TGRM08]. Most of these list-wise approaches directly optimize a performance measure in information retrieval, such as Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) [Liu09].

In the summarization task, there is no clear performance measure for the ranked sentence list. Note that the ranked sentence list is still an intermediate result for summarization and redundancy removal is needed to form the final summary. Hence, we develop our summarization system based on ranking SVM, a typical pairwise learning to rank method. Other pairwise learning to rank methods include RankBoost [FISS03] and RankNet [BSR$^+$05]. Our modification of ranking SVM is inspired by adopting cost sensitive loss function to differentiate document pairs from different queries or in different ranks [XCLH06, CXL$^+$06].

Most learning to rank methods, however, are based on the available high-quality training data. This is not the case when we apply these methods for summarization, where the training data needs to be automatically generated from the set of <query, document set, human summaries> triples.

5.2.2 Model Learning

Under the feature-based summarization framework, normally the scoring function needs to combine the impacts of various features. A common way is to use the linear combination of the features by tuning the weights of the features manually or empirically. The problem of such a method is that when the number of the features gets larger, the complexity of assigning weights grows exponentially. In this section, we explore the use of ranking SVM, a pairwise learning to rank model, for obtaining credible and controllable solutions for feature combinations.

Ranking SVM

Assume that a training set of labeled data is available. Given a training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ with $\mathbf{x}_i \in \Re^N$ and $y_i \in \{1, \ldots, R\}$. In the formulation of Herbrich et al. [HGO99], the goal is to learn a function $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, so that for any pair of examples $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ it holds that

$$h(\mathbf{x}_i) > h(\mathbf{x}_j) \Longleftrightarrow y_i > y_j.$$

In this way, the task of learning to rank is formulated as the problem of classification on pairs of instances. In particular, the SVM model can be applied and the task is thus formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, \xi_{ij} \geq 0} \quad & \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + \tfrac{C}{m} \sum_{(i,j) \in P} \xi_{ij} \\ s.t. \quad & \forall (i,j) \in P : \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j) > 1 - \xi_{ij}, \end{aligned} \tag{5.5}$$

where P is the set of pairs (i, j) for which example i has a higher rank than example j, i.e. $P = \{(i, j) : y_i > y_j\}$, $m = |P|$, and ξ_{ij}'s are slack variables. This optimization problem is equivalent to

$$\min_{\mathbf{w}} \frac{1}{2C}\mathbf{w}^T\mathbf{w} + \frac{1}{m} \sum_{(i,j) \in P} \max\{0, 1 - \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)\}, \tag{5.6}$$

	@1	@2	@3	@4	@5
Ranking 1(Perfect) :	s(.3)	s(.7)	n(.3)	n(.7)	n(.8)
Ranking 2(Perfect) :	s(.7)	s(.3)	n(.7)	n(.3)	n(.8)
Ranking 3 :	s(.3)	n(.7)	s(.7)	n(.3)	n(.8)
Ranking 4 :	s(.7)	n(.3)	s(.3)	n(.7)	n(.8)

Table 5.6: Example rankings for the five sentences.

where the second term is called "empirical hinge loss".

Cost Sensitive Loss

Since the rankings of the sentences in the training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ are estimated, applying the empirical hinge loss may not be proper. Let us consider the following example. Given five sentences $\{s(.3), s(.7), n(.3), n(.7), n(.8)\}$, where 's' and 'n' indicate two possible ranks: summary and non-summary respectively, and the value in parentheses indicates the confidence score of the rank. Table 5.6 shows four possible rankings for these five sentences. Ranking 1 and Ranking 2 are both perfect, since it does not matter to swap two positions of both non-summary sentences or both summary sentences. Apparently, neither Ranking 3 nor Ranking 4 is perfect, and without considering confidence, they have the same quality. However, Ranking 4 should be better than Ranking 3 if we take the confidence into consideration. For the pair $< n(.7), s(.7) >$ in Ranking 3, $n(.7)$ is likely to be a non-summary sentence, and $s(.7)$ is likely to be a summary sentence. Therefore, we have good confidence that their relative positions should be swapped. For the pair $< n(.3), s(.3) >$ in Ranking 4, $n(.3)$ is less likely to be a non-summary sentence and $s(.3)$ is less likely to be a summary sentence. Their relative positions may be correct while their ranks might be mislabeled.

To deal with this problem, we adopt the idea of sensitive cost loss for SVM, and use penalty weight σ_{ij} for the loss function of each sentence pair. So the optimization problem in Eq. (5.6) becomes

$$\min_{\mathbf{w}} \frac{1}{2C}\mathbf{w}^T\mathbf{w} + \frac{1}{m}\sum_{(i,j)\in P} \max\{0, \sigma_{ij}(1 - \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j))\}. \tag{5.7}$$

In our task, for the sentence pair $< \mathbf{x}_i, \mathbf{x}_j >$, the sum of confidence scores of $\mathbf{x}_i$ and $\mathbf{x}_j$ (represented by c_i and c_j, respectively) can be used as the penalty weights. In other words,

$$\sigma_{ij} = c_i + c_j.$$

Basically, a pair of a non-summary sentence and a summary sentence with small confidence having reversed relative ranking positions will be less penalized than those with high confidence. To solve the problem in Eq.(5.7), we can solve the equivalent problem

$$\begin{aligned} \min_{\mathbf{w},\xi_{ij}\geq 0} \quad & \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + \tfrac{C}{m}\sum_{(i,j)\in P}\xi_{ij} \\ s.t. \quad & \forall (i,j) \in P : \sigma_{ij}(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)) \geq \sigma_{ij} - \xi_{ij}. \end{aligned} \tag{5.8}$$

5.2.3 Training Data Construction: A Graph based Method

In order to apply learning to rank for summarization, we need to have the labeled training set in the form of $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i$ is a sentence and y_i is the ranking of the sentence. Given a set of triples <query, document set, manual summary set>, instead of manually labeling the rank for every sentence, which is a time-consuming task, we can estimate the rank of a sentence with the reference of the manual summaries. For simplicity, we only assign the sentence with two possible ranks: summary or non-summary[1].

Note that generally human summaries do not contain redundancy. Therefore, to construct the training data, the sentences that have similar meanings to sentences in human

[1]Since the ranks are estimated, more ranks may introduce more noise, and we show later on in our experiments that more ranks are not necessary.

summaries but of lexical diversity should also be labeled as summary sentences. So, instead of simply comparing sentences in the document set with those in human summaries, we take the training data construction as an extractive summarization task, where the similarity between sentences in the documents set are also considered, and similar sentences should have similar probabilities to be labeled as a summary sentence [OER05]. Different from a standard extractive summarization, here redundancy removal is performed and the human summaries are used as the query.

To estimate the probability score $p(s|H)$ of a sentence s being labeled as a summary sentence given the human summary set H, we measure its relevance with sentences in the human summary set and its similarities with the other sentences in the document set. Formally, $p(s|H)$ is computed by the following formula:

$$\begin{aligned} p(s|H) = \quad & d \sum_{v \in C} \frac{sim(s,v)}{\sum_{z \in C} sim(z,v)} p(v|H) \\ & + (1-d) \frac{rel(s,H)}{\sum_{z \in C} rel(z,H)}, \end{aligned} \tag{5.9}$$

where C is the set of all sentences in the document set, and d is a trade-off parameter in the interval [0, 1], used to specify the relative contribution of the two terms in Eq.(5.9). For bigger value of d, more importance is given to the sentence-to-sentence similarity compared to sentence-to-human-summary relevance. The denominators in both terms are used for normalization. The matrix form of Eq.(5.9) can be written as

$$p(k+1) = M^T p(k), \tag{5.10}$$

$$M = dA + (1-d)B, \tag{5.11}$$

where M, A, and B are all square matrices. Elements in A represent the similarities between sentences in the document set. All elements of i-th column in B are proportional to $rel(i|H)$. A and B are both normalized to make the sum of each row equal to 1. Note

that k represents the kth iteration, and $p = [p_1, \ldots, p_N]^T$ is the vector of sentence ranking scores that we are looking for, which corresponds to the stationary distribution of the matrix M. The iteration is guaranteed to converge to a unique stationary distribution given that M is a stochastic matrix. To calculate the similarity of sentences in the document set, we use the cosine similarity. To calculate $rel(s, H)$, the sentence relevance given the human summary set, we use

$$rel(s, H) = \max_{r \in H} \text{ROUGE-2}_r(s), \tag{5.12}$$

where r is a sentence in the human summary, and ROUGE-$2_r(s)$ is the ROUGE-2 score of the sentence s with the reference r.

After estimating the score of every sentence in document set, a threshold is applied to assign a sentence rank 1 indicating summary sentence if the score is larger than the threshold, or otherwise rank 0 indicating non-summary sentence. The confidence score can be defined as

$$c_i = |p(x_i|H) - \text{threshold}|. \tag{5.13}$$

5.2.4 Feature Design

In our work, we use some common features that are widely used in the supervised summarization methods [OLL07, SSL$^+$07] as well as several features induced from the unsupervised methods for learning the model. In total 20 features are used in our work.

Basic Features

The basic features are the commonly used features in previous summarization approaches, which can be extracted directly without complicated computation. Given a query and

sentence pair, $< q, x_i >$, the basic features used for learning are described as follows.

Position: The position feature, denoted by Pos, indicates the position of x_i along the sentence sequence of a document. If x_i appears at the beginning of the document, Pos is set to be 1; if it is at the end of the document, Pos is 2; Otherwise, Pos is set to be 3.

Length: The length feature is the number of terms contained in x_i after removing the stop words according to a stop word list.

Number of Frequent Thematic Words: Thematic words are the most frequent words appeared in the documents after removing the stop words. Sentences containing more thematic words are more likely to be summary sentences. We use the number of frequent thematic words in x_i as a feature. In our work, 5 frequency thresholds 10,20,50,100, 200 are used to define the frequent thematic words, thus generating 5 features for each sentence.

Similarity to the Closest Neighboring Sentences: We also use the average similarity between a sentence and its closest neighbors as features. In particular, we use "Intra Sim to Pre N" and "Intra Sim to Next N" (N = 1, 2, 5) to record the average similarity of x_i to the previous N most similar sentences and to the next N most similar sentences respectively, in the same document. "Inter Sim to N" (N = 1,2,5) is also used to record the average similarity of x_i to the N most similar sentences in different documents. We use the cosine measure to compute the similarity measurement.

Similarity to the Query: The cosine similarity between the query q and the sentence x_i is also used as a feature.

Complex Features

Manifold Ranking Score: The ranking score is obtained for each sentence in the manifold-ranking process to denote the biased information richness of the sentence. All sentences in the document set plus the query description are considered as points $\{x_0, x_1, \ldots, x_n\}$

in a manifold space, where x_0 is the query description and the others are the sentences in the documents. The ranking function is denoted by $f = [f_0, f_1, \ldots, f_n]$. Since x_0 is the query description, the initial label vector of these sentences is $y = [y_0, y_1, \ldots, y_n]$, where $y_0 = 1, y_1 = \ldots = y_n = 0$. The manifold ranking can be computed iteratively using the following equation,

$$f(k+1) = \alpha S f(k) + (1-\alpha)y, \tag{5.14}$$

where S is the symmetrically normalized similarity matrix of $\{x_0, x_1, \ldots, x_n\}$, and α is a parameter, and k represents the k-th iteration. The iterative algorithm is guaranteed to converge to the final manifold ranking scores [WYX07a]. We set the α to 0.3, 0.5, 0.8 to obtain three different manifold ranking scores as three features. More detailed description of manifold ranking score can be found in [WYX07a].

Redundancy Removal

To generate the final summary, all our implemented methods use the diversity penalty algorithm as in [WYX07a] to impose redundancy penalty. as described in Algorithm 3. At each iteration of line 3-7, the sentence with the maximum score is selected into the summary, and other sentences are penalized according to their similarities to the selected sentence. A in line 7 indicates the normalized similarity matrix of all sentences.

Algorithm 3 Generate Final Summary

Require: sentence set: $S_1 = \{s_1, \ldots, s_n\}$,
scoring function: $f(s_i), 1 \leq i \leq n$,
Ensure: Summary: S_2

1: Initialize $S_2 = \emptyset$, score$(s_i) = f(s_i)$
2: **while** $S_1 \neq \emptyset$ and S_2 does not reach limit **do**
3: $s_{i^*} = \arg\max_{s \in S_1}$ score(s)
4: $S_1 = S_1 - \{s_{i^*}\}$
5: $S_2 = S_2 \cup \{s_{i^*}\}$
6: **for** s_j in S_1 **do**
7: score(s_j) = score$(s_j) - A_{ji^*} f(s_{i^*})$

5.2.5 Experiments

Experiment Settings

We evaluate our proposed method for query-focused multi-document summarization on the main tasks of DUC 2005, DUC 2006, and DUC 2007. Each task has a gold standard data set consisting of document sets and reference summaries. In our experiments, DUC 2005 is used to train the model tested on DUC 2006, and DUC 2006 is used train the model tested on DUC 2007. Table 5.7 lists the characteristics of the data sets.

	DUC 2005	DUC 2006	DUC 2007
#topics	50	50	45
#documents per topic	25-50	25	25
Summary length	250 words	250 words	250 words

Table 5.7: Brief description of the data sets.

We use ROUGE toolkit (version 1.5.5) [LH03], described in Section 5.1.4, to measure the summarization performance.

In the following experiments, we use ROUGE-1, ROUGE-2, ROUGE-W and ROUGE-SU, of which ROUGE-2 and ROUGE-SU were adopted by DUC 2006 and DUC 2007 for automatic performance evaluation, and all of which are widely used in summarization research.

SVM^{Rank} [Joa06] is used as a tool for ranking SVM and also served as a basis for ranking SVM with cost sensitive loss. The parameter C in Eq.(5.5) and Eq.(5.8) is set to 1 for all following experiments, and other parameters are set to the default values. The threshold for assigning the two ranks to the sentences in training data generation is chosen by 10-fold cross validation.

	ROUGE-1	ROUGE-2	ROUGE-SU
Ranking-SVM-CSL	.4221 (.4158-.4279)	.0994 (.0949-.1034)	.1542 (.1503-.1579)
Ranking-SVM	.4215 (.4155-.4275)	.0983 (.0942-.1026)	.1533 (.1495-.1560)
SVR	.4166 (.4104-.4226)	.0952 (.0912-.0992)	.1517 (.1480-.1555)
Manifold-Ranking	.3882 (.3821-.3944)	.0801 (.0761-.0842)	.1370 (.1333-.1409)
S24	.4111 (.4049-.4171)	.0951 (.0909-.0991)	.1547 (.1506-.1584)
S12	.4048 (.3992-.4105)	.0899 (.0858-.0939)	.1475 (.1436-.1514)
S23	.4044 (.3982-.4097)	.0879 (.0837-.0920)	.1449 (.1410-.1485)

Table 5.8: Summarization performance comparison on DUC 2006.

	ROUGE-1	ROUGE-2	ROUGE-SU
Ranking-SVM-CSL	.4496 (.4435-.4557)	.1229 (.1182-.1270)	.1710 (.1665-.1758)
Ranking-SVM	.4461 (.4396-.4526)	.1203 (.1158-.1247)	.1701 (.1658-.1742)
SVR	.4395 (.4329-.4466)	.1179 (.1132-.1224)	.1652 (.1607-.1696)
Manifold-Ranking	.3957 (.3899-.4022)	.0769 (.0733-.0809)	.1362 (.1329-.1400)
S15	.4451 (.4379-.4521)	.1245 (.1196-.1293)	.1771 (.1724-.1818)
S29	.4325 (.4260-.4387)	.1203 (.1155-.1253)	.1707 (.1609-.1806)
S4	.4342 (.4291-.4391)	.1189 (.1146-.1237)	.1700 (.1661-.1754)

Table 5.9: Summarization performance comparison on DUC 2007.

System Comparison

First we compare our method Ranking-SVM-CSL (Ranking SVM with Cost Sensitive Loss) with three competitive baselines and three top systems of DUC. The baseline systems include 1) Ranking-SVM: applying ranking SVM directly; 2) SVR: learning a regression model using SVM; and 3) Manifold-Ranking: ranking the sentences according to the manifold ranking score, which is one of the features described in the previous section, where the parameter α is set to 0.5. All of the three baselines use the proposed graph based method in training data generation. The top three systems are the three systems with highest ROUGE-2 scores, chosen from the participant systems of DUC 2006 and DUC 2007, respectively, and are represented by their system IDs.

Table 5.8 and Table 5.9 present the performance of these systems in ROUGE-1, ROUGE-2, ROUGE-W and ROUGE-SU along with corresponding 95% confidence in-

tervals. As in [NVM06], we approximately determine which differences in scores are significant via comparing the 95% confidence intervals, and significant differences are those where the confidence intervals for the estimates of the means for the two systems either do not overlap at all, or where the two intervals overlap but neither contains the best estimate for the mean of the other. From the results we can observe that our proposed method outperforms all baseline systems, performs significantly better than S12 and S23 on DUC 2006 and comparative to the two top systems S24 and S15 on DUC 2006 and DUC 2007 respectively, in most of ROUGE measures. It should be pointed out that the top systems in DUC involves much more preprocessing and postprocessing such as sentence reduction and entity de-referencing in S15 [PRV07].

Manifold-Ranking has the worst performance since it only uses the manifold ranking score as the single feature. Combination of multiple features leads to a significant improvement. Among the systems that automatically learn the combination weights for various features, learning to rank based methods (Ranking-SVM-CSL and Ranking-SVM) outperform the regression model (SVR). In particular, Ranking-SVM-CSL improves SVR significantly in respect of ROUGE-W on the DUC 2006 dataset and all except ROUGE-2 on the DUC 2007 dataset, while Ranking-SVM improves SVR significantly only in respect of ROUGE-W on both datasets. Note that standard learning to rank methods focus on the ranking of the sentences and do not use the scores of the sentences. With Ranking-SVM-CSL, the scores of sentences are used as confidence in the loss function for sentence pairs, which leads to better performance than directly applying ranking SVM.

Training Data Generation Comparison

In this section, we empirically investigate the effects of different strategies for training data generation. We denote the proposed method of training data construction as graph-based-method and compare it with a set of baselines described below.

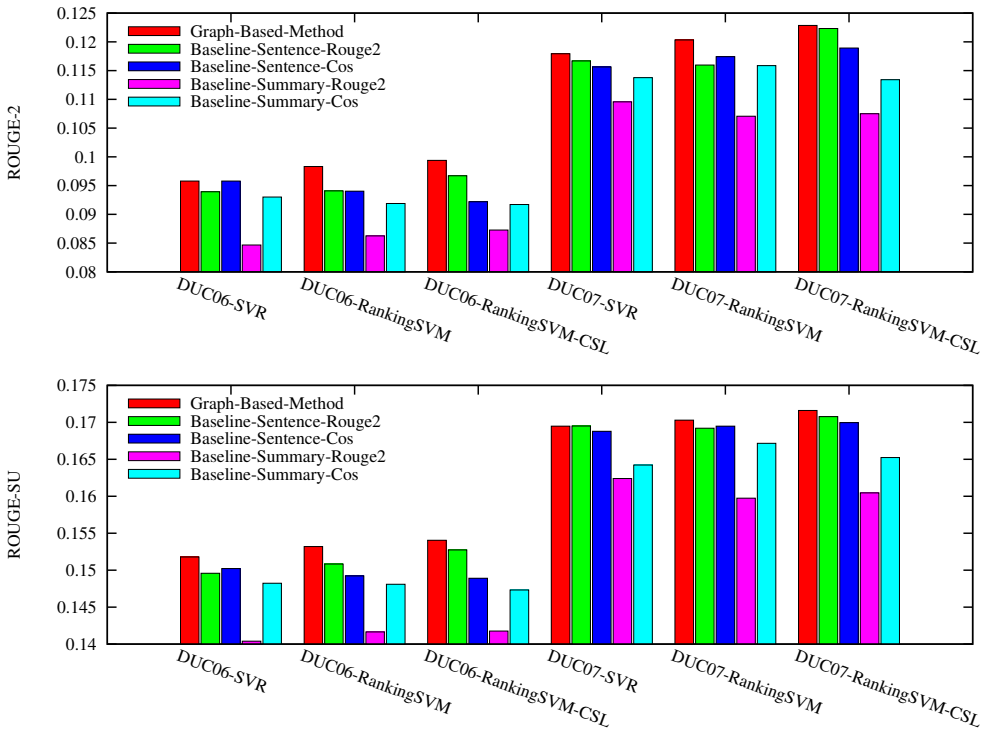


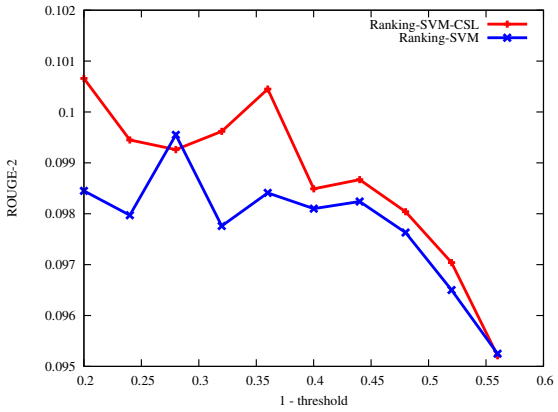
Figure 5.4: Performance comparison of training data generation.

Given a summary set H for a query and a set of sentences $\{x_i\}_{i=1}^{N}$ in a set of documents, generally, the following strategy can be used to estimate the ranks of the sentences:

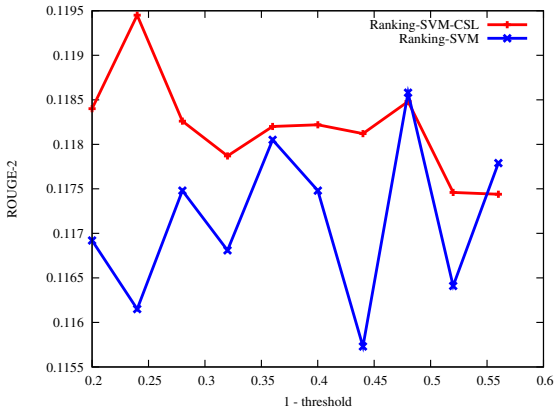
$$y_i^* = \max_{e \in H} y_{i,e}^* \tag{5.15}$$

where y_i^* is the estimated rank of sentence i, e is the reference which can be a sentence or a summary in H, $y_{i,e}^*$ is a discretized result of $sim(x_i, e)$ where sim can be the cosine similarity or ROUGE score of the sentence given the reference, representing the probability x_i is summary given the reference e.

We compare our graph-based method to this baseline strategy with different references (sentence or summary) and different similarity measurements (cosine similarity or ROUGE-2 score) and the comparison is shown in Figure 5.4. From the comparison, we observe that: 1) Using sentence as the reference is much better than using the whole summary, especially with the ROUGE score as the similarity function. This may due



(a) DUC 2006



(b) DUC 2007

Figure 5.5: Effects using cost sensitive loss. (Value of x-axis represents $1 - \text{threshold}$)

to the fact that more different words in the whole summary may lead to a bias in favor of those longer sentences having more overlapping grams with the reference, especially using similarity functions with no normalization factor, like ROUGE-2 score. 2) Our graph-based method outperforms other baseline strategies in most of combination of data and learning models. This is because our graph-based method makes use of the sentence relationships in the documents set, which has been shown as an important factor in a lot of summarization work to score the sentences.

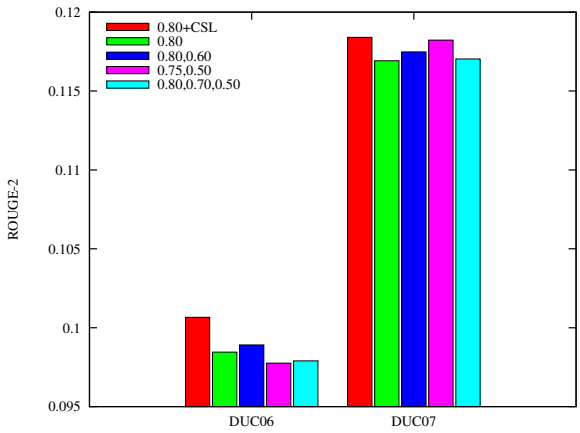


Figure 5.6: Performance comparison using training data with multiple ranks.

Effect of Cost Sensitive Loss

In this section, we empirically investigate the effect of the cost sensitive loss. Figure 5.5(a) and Figure 5.5(b) show the performance comparison between Rank-SVM-CSL (with cost sensitive loss) and Ranking-SVM (without cost sensitive loss) for different thresholds on DUC 2006 and DUC 2007, respectively. For most thresholds we test, cost sensitive loss improves the performance on both DUC 2006 and DUC 2007. We can observe that the performance of Ranking SVM, especially in Figure 5.5(b) changes frequently with the variation of the threshold. Compared with directly using ranking SVM, the results of Ranking-SVM-CSL are more stable.

Granularity of Rank

In our work, the sentences of the document set are divided into two ranks: summary and non-summary. Here we use a case study to show that more ranks do not lead to significant performance improvements. Instead of using only one threshold (0.8 in this case), we map the sentences to more than two ranks by selecting more than one thresholds. Intuitively, the number of summary sentences should be less than the number of non-

summary sentences. Hence the thresholds are chosen to make the number of sentences in a higher rank less than that in a lower rank.

Figure 5.6 shows the performance using ranking SVM using different thresholds. "+CSL" indicates learning with ranking SVM with cost sensitive loss. We observe that: although using 3 or more ranks (i.e., with 2 or more thresholds) may lead to better results (e.g., (0.80,0.60) on DUC 2006 and DUC 2007, (0.75,0.50) on DUC 2007, and (0.80,0.70,0.50) on DUC 2007), the improvement is unstable and small, compared with the improvement made by 0.80+CSL (i.e., using threshold 0.8 followed by learning with ranking SVM with cost sensitive loss). We leave it as future work to explore the effects of applying cost sensitive loss to cases with more than two ranks.

5.3 Summary

In this chapter, we propose two frameworks for multi-document summarization for flexible information needs. The first framework models multi-document summarization using the minimum dominating set, and shows it versatility to formulate many well-known summarization tasks with simple and effective summarization methods. The second framework incorporates a learning to rank approach, ranking SVM, to combine features for extractive query-focus multi-document summarization. To apply ranking SVM for summarization, we propose a graph-based method for training data generation by utilizing the sentence relationships and introduce a cost sensitive loss to improve the robustness of learning.

CHAPTER 6

APPLICATION: EVENT SUMMARIZATION FOR SPORTS GAMES USING TWITTER STREAMS

6.1 Introduction

Thousands of events are being discussed on the social media websites everyday. Using the social media, people report the events they are experiencing or publish comments on the events in real-time, which are aggregated into a highly valuable stream of information that informs us the events happening around the world. But on the other hand, the large number of posts from millions of social media users often leads to the information overload problem. Those who search for the information related to a particular event often find difficulty to get a big picture of it, given the overwhelmingly large collection of data.

Event summarization aims to provide a textual description of an event of interest to address this problem. Given a data stream consisting of chronologically-ordered text pieces related to an event, an event summarization system aims to generate an informative textual description that can capture all the important moments and ideally the summary should be produced in a progressive manner as the event unfolds.

Among these events, the sports games receive a lot of attention from the Twitter audience. In this chapter, we present a novel participant-centered event summarization application for sports games using the Twitter stream. The application provides an alternative way to be kept informed of the progress of a sports game and audience's responds from the social media data. The summary of the progress of a game can be delivered in real-time to the sports fans who cannot make it to the game or watch it at home; the automatically generated summary can also be supplied to the news reporters to assist with the writing of the game recap which provides a full coverage of the exciting moments happened on the playground.

To build the application, aforementioned text analysis methods on social media are integrated. For a game, we first get a filtered Twitter stream using a set of keywords including names of teams, players and coaches. Then the *participant-based event detection* is applied on the event stream data to detect the important moments during the event, a.k.a sub-events. The *dominating set based summarization approach* is then applied to the multiple tweets of each sub-event. Besides a summary, we also utilize a *sentiment classifier* to automatically classify a tweet into one of the three categories "positive","negative" and "neutral" to reflect the game audience's emotion change during the game.

6.2 Framework Overview

We propose a novel participant-centered event summarization approach that consists of three key components: (1) "Participant Detection" dynamically identifies the event participants and divides the entire event stream into a number of participant streams ; (2) "Sub-event Detection" introduces a novel time-content mixture model approach to identify the important sub-events associated with each participant; these "participant-level sub-events" are then merged along the timeline to form a set of "global sub-events"[1], which capture all the important moments in the event stream; (3) "Summary Tweet Extraction" extracts the representative tweets from the global sub-events and forms a comprehensive coverage of the event progress.

In Figure 6.1, we provide an overview of the system framework. It consists of three main components: sub-event detection, participant detection and summary generation.

[1]We use "**participant sub-events**" and "**global sub-events**" respectively to represent the important moments happened on the participant-level and on the entire event-level. A "global sub-event" may consist of one or more "participant sub-events". For example., the "steal" action in the basketball game typically involves both the defensive and offensive players, and can be generated by merging the two participant-level sub-events.

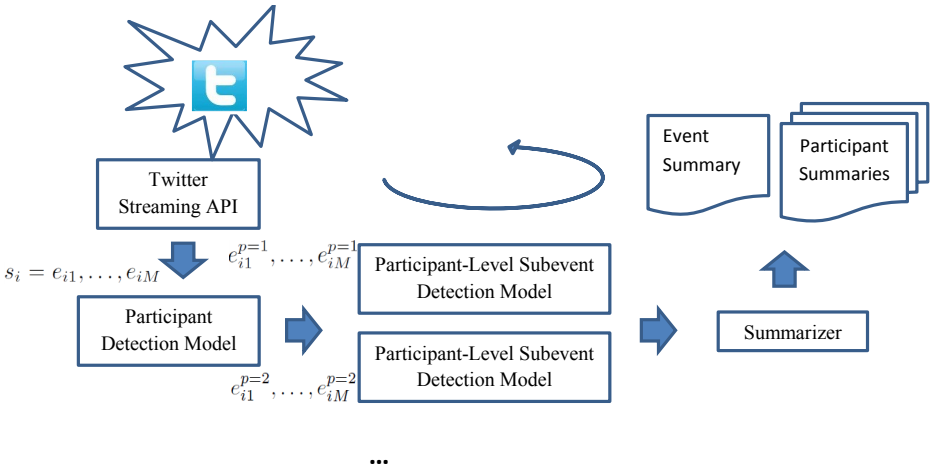


Figure 6.1: System framework of the event summarization application for sports games using Twitter streams.

To collect the stream of tweets about a particular event, the system requires users to input the start and end time of the event, and a set of keywords, and calls Twitter's streaming APIs the obtain tweets containing one of the keywords during the event's time period.

- **Participant Detection**: The goal of participant detection is to identify the important entities in the stream that play a significant role in shaping the event progress. We introduce an online clustering approach to automatically group the mentions referred to the same entities in the stream, and update the model for every input segment of tweets s_i. According to the clustering results, the input segment can be devided into several sub-segments, one for each participant p, as s_i^p, composed of those tweets of s_i containing a mention of the participant p.
- **Sub-event Detection**: Given a participant stream, the proposed sub-event detection algorithm automatically identify the important moments (a.k.a. sub-events) in the stream based on both content salience and the temporal burstiness of the stream. Each sub-event is represented by a set of associated tweets and a peak time, when the tweet volume has reached a peak during that time period.

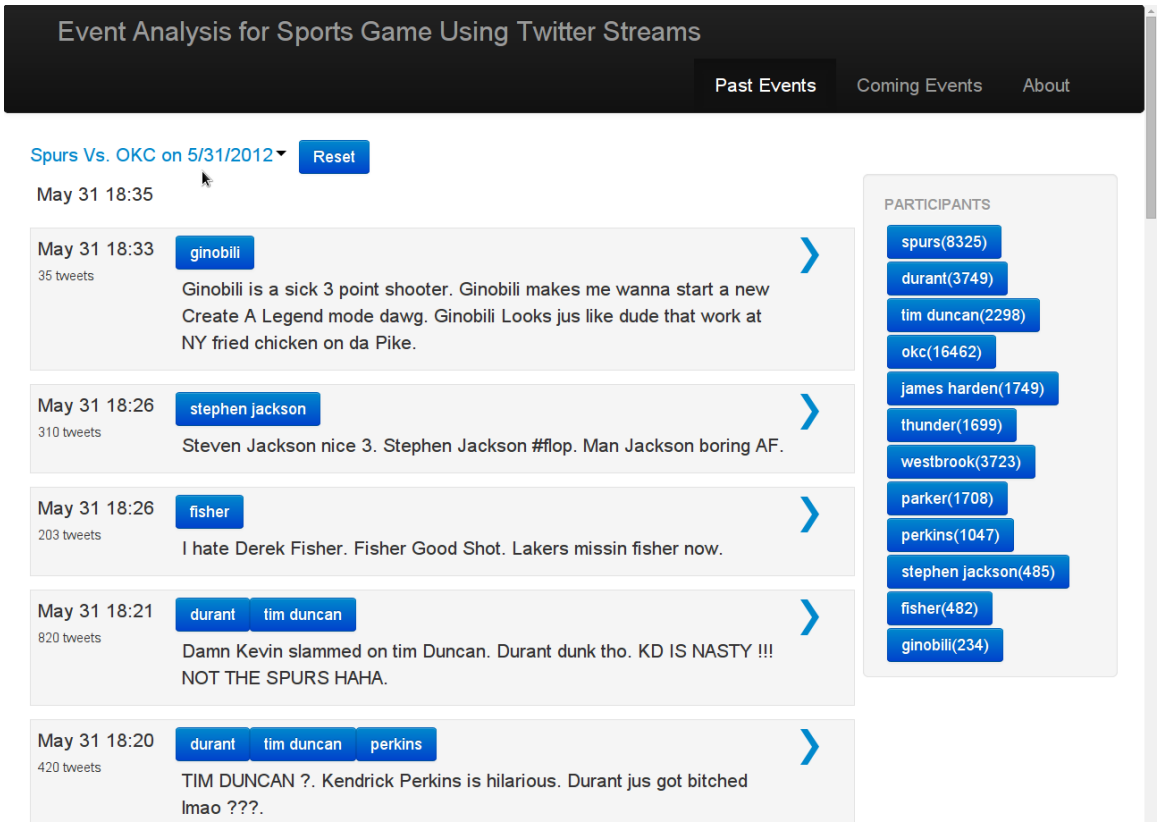


Figure 6.2: Screenshot of the sub-event list of the system.

- **Summary Generation**: The summary generation module takes the input of sets of tweets, each associated with a sub-events of a participant, and aims to generate a high-quality textual summary as well as a sentiment summary.

In an online framework, each of these key components, including the sub-event detection, participant detection, and summary generation, maintains a set of parameters and they are constantly updated when a new segment of tweets become available.

Figure 6.2 and Figure 6.3 show screen-shots of our system. In Figure 6.2, users can choose to replay a previous event or follow a current ongoing event. As the related tweets of the chosen event are being fed into the system, filtered by predefined keywords related with the event, new sub-events are detected and summarized automatically, and inserted into the top of the main part of the page. The right side of the page lists participants

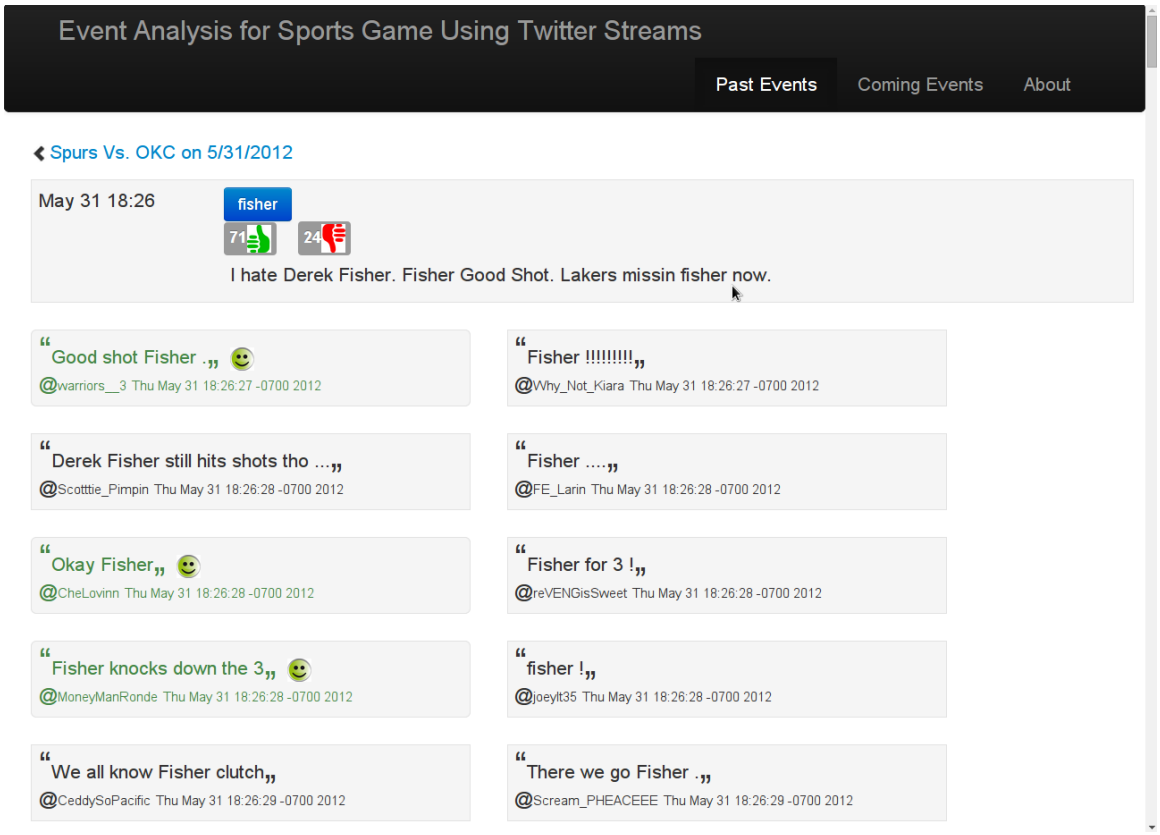


Figure 6.3: Screenshot of the sub-event details of the system.

of the event. The number by each participant indicates the number of tweets where this participant is discussed, by which users can find the most popular participants so far. To obtain more information about a participant users are interested in, they can furthur zoom-in to a particular participant to list all the sub-events the participant is involved in so far. After users click the arrow icon beside a sub-event summary in Figure 6.2, they go to a detailed page of the sub-event as shown in Figure 6.3, including the list of all tweets about the sub-events and a sentimental analysis result. For showing the aggregated sentiment of Twitter users for each sub-event, the system calculates the numbers of positive and negative tweets of the sub-event respectively, after conducting a sentiment classification on each tweet.

6.3 Online Participant Detection

For the online requirement, we formulate the participant detection as an incremental cross-tweets co-reference resolution task in a twitter stream. A named entity recognition tool [RCME11] is used for named entity tagging in tweets. Then the tagged named entities (a.k.a., mentions) are grouped into clusters using a streaming clustering algorithm, which consists of two stages: update and merge, applied to each new incoming segment of tweets. Update adds mentions to existing clusters if the similarity between the mention and an existing cluster is less than a threshold δ_u or otherwise creates new clusters, while merge itself is hierarchical agglomerative clustering to revise the clustering result by combining them.

In the update stage, we define the similarity of a mention m and an existing cluster c as

$$sim(m, c) = \alpha lex(m, c) + (1 - \alpha) context(m, c), \tag{6.1}$$

where $lex(m, c)$ captures lexical resemblance between m and mentions in c and and $context(m, c)$ cosine similarity between contexts of m and c. $lex(m, c)$ can be calculated as portion of overlapping n-grams between them as

$$lex(m, c) = \frac{|ngram(m) \cap ngram(c)|}{|ngram(m) \cup ngram(c)|}. \tag{6.2}$$

For example, in the following two tweets "*Gotta respect Anthony Davis, still rocking the unibrow*", "*Anthony gotta do something about that unibrow*", the two mentions *Anthony Davis* and *Anthony* are referring to the same participant and they share both character overlap ("anthony") and context words ("unibrow", "gotta"). However, for mentions in tweets, their context information is very limited and may vary a lot even they referred to the same entity. The previous update process may lead to a large of number of new clusters which lower efficiency of the system. Instead of updating the clustering by one mention each time, by assuming that mentions in one segment with same name refer to

the same entity, we first group all mentions with the same name in the segment, extract context for the mentions and select a cluster to assign all these mentions to.

To further reduce the cluster number, since participants we want to detect are entities that play significant roles, we can discard some infrequent entities. For a name, if there are more than δ_l continuous slices in each of which there are more than δ_s mentions of name, we activate the name. So we only keep track of mentions with frequent names.

In the merge stage, a hierarchical agglomerative clustering is conducted with a stopping threshold δ_m. Since we suppose to have sufficient context information in this stage and our goal to combine mentions with different names, here only context similarity is used to measure the similarity between clusters while lexical resemblance is used as constraints. To combine two clusters, at least half of mentions in both clusters needs to be lexically related with a mention in each other. A mention m is lexically related with mention m' if $m(m')$ is an abbreviation, acronym, or part of another mention $m'(m)$, or or if the character edit distance between the two mentions is less than a threshold θ[2].

6.4 Online Update for a Temporal-Content Mixture Model

When we have all the tweets about the event, EM algorithm can be applied to the whole data to train the event detection model, as proposed in Chapter 4. However, in real case, we are more interesting in summarizing the on-going event in real-time.

To process a data stream D, we first split it into 10-second time slices $D = s_1, s_2, \ldots$. Each slice contains a set of tweets that were published during that time interval.

In an online processing mode using the same temporal-content mixture model, the system iteratively consumes the new w_{new} slices of tweets each time to update the model parameters with the most recent $w_{working}$ slices of tweets in memory. The $w_{working}$ slices

[2] θ was empirically set as $0.2 \times min\{|m|, |m'|\}$

can be further divided into updating area, fixed area in Figure 6.4, where a Gaussian distribution is used to represent a sub-event topic.

Due to the locality of a sub-event, we assume independency between the sub-events before updating area (including reserved and fixed area) and the incoming tweets, so that only parameters for those sub-event topics in the updating area are updated with new incoming tweets. For the same reason, the oldest tweets in the fixed area are least likely to belong to a much older sub-event topic, so we only need to keep the parameters of the sub-event topics in reserved area in memory. In the application, we set 10min for width of the updating area, 15min for width of the reserved area, and 5min for the fixed area to keep tweets of 20min in memory.

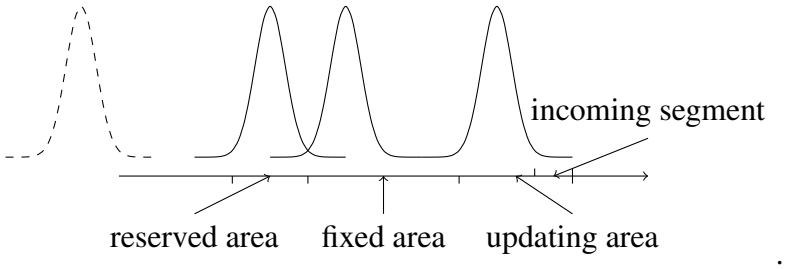


Figure 6.4: Illustration of how sub-events are detected online.

A data segment is represented as w slices: $D_i = s_i, s_{i+1}, \ldots, s_{i+w-1}$. We use K and B to denote the number of sub-event topics and background topics currently contained in the model. B was empirically set to 2 initially. The following steps are repeated to process each data segment:

EM Initialization When a new data segment D_i becomes available, we need to update the number of sub-event topics ΔK and a background topic ΔB, as well as re-initialize the model parameters (μ, σ, θ) for both sub-event and background topics. Initially we set the increment of the sub-event topics empirically ($\Delta K = 1$) and keep the number of background topics unchanged ($\Delta B = 0$). Later we will perform a topic readjustment

process to further adjust their numbers. For the new sub-event topics, its Gaussian parameters μ and σ are initialized using the tweets in the new data segment; its multinomial parameters are initialized randomly. The new data segment D_i also introduces unseen words which we use to expand our existing vocabulary. For both existing sub-event topics and background topics, the multinomial parameters corresponding to these new words are initiated randomly to a small value.

EM Update To perform the EM update, we only involve the sub-event topics that are most close to the current time point in the new EM update process. They are the ones whose peak time $\hat{t}$ is within updating area. Their parameters will likely be changed given a new segment of the data stream. The parameters of the earlier sub-event topics are fixed and will not be changed anymore. In addition, we would like to involve only the most recent tweets in the model update. We use only those tweets who are published in fixed area and updating area. Those tweets that are published earlier are discarded. These tweets are used together with the new data segment for the new EM update.

EM Postprocessing A topic re-adjustment was performed after the EM process. We merge two sub-events in a data stream if they (1) locate closely in the timeline, with peaks times within a 2-minute window, where peak time of a sub-event is defined as the slice that has the most tweets associated with this sub-event; and (2) share similar word distributions if their symmetric KL divergence is less than a threshold ($thresh_{sim} = 5$). We also convert the sub-event topics to background topics if their σ values are greater than a threshold β[3]. We then re-run the EM process to obtain the updated parameters. The topic re-adjustment process continues until the number of sub-events and background topics do

[3] β was set to 5 minutes in our experiments.

not change further. We only output the sub-event topic is the number of associated tweets in its peak time is larger than a threshold (=15).

We obtain the "**participant sub-events**" by applying this sub-event detection approach to each of the participant streams. The "**global sub-events**" are obtained by merging the participant sub-events along the timeline. We merge two participant sub-events into a global sub-event if (1) their peaks are within a 2-minute window, and (2) the Jaccard similarity [L.99] between their associated tweets is greater than a threshold (set to 0.1 empirically). The tweets associated with each global sub-event are the ones with $p(z|d)$ greater than a threshold γ, where z is one of the participant sub-events and γ was set to 0.7 empirically. After the sub-event detection process, we obtain a set of global sub-events and their associated event tweets.[4]

6.5 Experiments

Similar in Chapter 4, we evaluate the proposed event summarization application on five NBA basketball games[5] as shown in Table 6.1.

	Event	Date	Duration	#Tweets
	Lakers vs Okc	05/19/2012	3h10m	218,313
N	Celtics vs 76ers	05/23/2012	3h30m	245,734
B	Celtics vs Heat	05/30/2012	3h30m	345,335
A	Spurs vs Okc	05/31/2012	3h	254,670
	Heat vs Okc	06/21/2012	3h30m	332,223

Table 6.1: Statistics of the data set, including five NBA basketball games event.

[4]We empirically set some threshold values in the topic re-adjustment and sub-event merging process. In future, we would like to explore more principled way of parameter selection.

[5]We remove the game event Heat vs OKC on 06/12/2012, which is almost duplicated with Heat vs OKC on 06/21/2012, comparing with the datasets used in Chapter 4.

6.5.1 Participant Detection

We evaluate the participant detection similar as a cross-tweet co-reference solution task. To build labeled co-reference data, for every event, we first sample hundreds to over a thousand tweets containing one of 50 most frequent names in the event; then an annotator labeled these sampled tweets with chains of entities. Singletons and those mentions which are not referred to an actually participant of the event (e.g., "Kevin" referred to a cousin of the tweet author, or "Jessica" referred to a performer on American Idols). B-Cubed [BB98], is most widely used in co-reference resolution evaluation, is used as the metric compare participant detection result and the labeled data. Recall score of B-Cubed is calculated as:

$$B_R^3 = \frac{1}{N} \sum_{d \in D} \sum_{m \in d} \frac{O_m}{S_m} \tag{6.3}$$

where D, d and m are the set of documents, a document, and a mention, respectively. S_m is the set of mentions of the annotated mention chain which contains m, while O_m is the overlap of S_m and the set of mentions of the system generated mention chain which contains m. N is the total number of mentions in D. The precision is computed by switching the role of annotated data and system generated data. F-measure is computed as geometrical average of recall and precision.

We evaluate the participant detection method used in the application system, referred to as SegmentUpdate, by comparing it with following baselines:

ExactMatch The method which clusters mentions only based on names.

TweetUpdate In update stage, clustering is updated once for a mention in a tweet.

IncNameHAC It is an incremental version of NameHAC, updating the hierarchical tree based on the available part of the stream, by conducting further merge.

NameHAC Hierarchical agglomerative clustering on names of mentions, assuming mentions with the same name refer to the same entity. For a pair of names, their similarity is

Approach	Lakers vs Okc			Celtics Vs 76ers			Celtics vs Heat		
	P	R	F	P	R	F	P	R	F
ExactMatch	0.981	0.692	0.811	0.825	0.585	0.685	0.893	0.696	0.782
TweetUpdate	1.000	0.658	0.794	0.913	0.660	0.766	0.847	0.720	0.779
IncNameHAC	1.000	0.542	0.703	0.820	0.589	0.686	0.822	0.650	0.726
SegmentUpdate	1.000	0.682	0.811	0.851	0.707	0.772	0.801	0.855	0.827
NameHAC	1.000	0.791	0.883	0.875	0.716	0.788	0.8884	0.918	0.903
	spursvsokc			heatvsokc					
	P	R	F	P	R	F			
ExactMatch	0.857	0.616	0.717	0.922	0.626	0.746			
TweetUpdate	0.877	0.712	0.786	0.952	0.712	0.815			
IncNameHAC	0.864	0.545	0.669	0.932	0.753	0.833			
SegmentUpdate	0.839	0.764	0.800	0.911	0.847	0.878			
NameHAC	0.853	0.774	0.811	0.948	0.843	0.892			

Table 6.2: Performance comparison of methods for participant detection.

based on the whole stream, so it is not applicable to our case, but can be seen as an upper bound.

Table 6.2 shows the comparing results. We can observe that 1) NameHAC has the best performance since it makes use of the whole data instead of conducting detection incrementally; 2) The incremental version of NameHAC does not perform well, even worse than the trivial method ExactMatch; 3) SegmentUpdate, which is used the application system, has a reasonable performance. It outperforms IncNameHAC since it allow two mentions composed of the same phrase refer to different participants, if the phrase is ambiguous. It also performs better than TweetUpdate, since it collects more information in phrase clustering for each phrase, from a segment of tweets instead of a single tweet.

6.5.2 Event Summarization

For each game, an annotator manually labels the sub-events according the play-by-play data from ESPN[6], and for each sub-event, representative tweets are extracted up to 140 characters as the manual summary.

[6]http://espn.go.com/nba/scoreboard

To evaluate the final summaries of an event, we following the work in [TYO11] to evaluate summarization for a document stream using a modified version of ROUGE [Lin04] score, which widely used as automatic evaluation for document summarization tasks. ROUGE measures the quality of a summary by counting the unit overlaps between the candidate summary and a set of reference summaries. Several automatic evaluation methods are implemented in ROUGE, such as ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-SU. ROUGE-N is an n-gram recall computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{ref}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{ref}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)}, \tag{6.4}$$

where n is length of the n-gram, ref stands for the reference summaries, $\text{Count}_{\text{match}}(\text{gram}_n)$ is the number of co-occurring n-grams in a candidate summary and the reference summaries, and $\text{Count}(\text{gram}_n)$ is the number of n-grams in the reference summaries. ROUGE-L uses the longest common sub-sequence (LCS) statistics, while ROUGE-W is based on weighted LCS and ROUGE-SU is based on skip-bigram plus unigram. Each of these evaluation methods in ROUGE can generate three scores (recall, precision and F-measure). However, ROUGE score cannot be applied directly to summarization of a document stream, in our case, a tweet stream about an event, since same n-grams that appear at distant time points describe different sub-events and should be regarded as different n-grams. In our manually labeled and system generated summaries, each n-gram is associated with the timestamp as the same of the sub-event the n-gram describes. Making use of such temporal information, we modify ROUGE-N to ROUGE^T-N, calculated as

$$\text{ROUGE}^T\text{-N} = \frac{\sum_{S \in \text{ref}} \sum_{\text{gram}_n^t \in S} \text{Count}_{\text{match}^T}(\text{gram}_n^t)}{\sum_{S \in \text{ref}} \sum_{\text{gram}_n^t \in S} \text{Count}(\text{gram}_n^t)} \tag{6.5}$$

where gram_n^t is a unique n-gram with a timestamp, and $\text{Count}_{\text{match}^T}(\text{gram}_n^t)$ returns the minimum of occurrence of n-gram with timestamp t in S and the number of matched n-grams in a candidate summary. The distance between the timestamp of a matched n-gram and t needs to be within a constant, which set to 1 min in our experiments.

Methods	Celtics Vs 76ers	Celtics vs Heat	Heat Vs Okc	Lakers vs Okc	Spurs Vs Okc
Spike	.2664	.31651	.2736	.2838	.2409
+Participant	.3240	.38784	.3016	.3399	.2917
MM	.3199	.38591	.3286	.3526	.2841
+Participant	.3571	.40162	.3493	.3899	.3063
MMOnline +Participant	.3428	.3970	.3163	.3852	.3068

Table 6.3: ROUGE$^{\mathrm{T}}$-1 F-1 scores

We compare the sub-event detection method used in the application system, referred to as MixtureModelOnline+Participant to the spike detection method (Spike) [MBB^{+}11] and the method batch-mode (MM) proposed in Chapter 4 based or not based on participant detection results. Table 6.3 shows the summarization evaluation results for comparing sub-event detection methods in term of the new evaluation metric ROUGE$^{\mathrm{T}} - 1$ F-1 score. From Table 6.3, we have several observations: 1) sub-event detection conducted based on participant streams leads to better summarization performance due to more accurate sub-event detection results; 2) The temporal-content mixture model outperforms the spike detection since the former takes the tweet content into consideration; 3) The online version of temporal-content mixture model, MMOnline+Participant, under-performs its batch counterpart, but their F-1 scores are close, which indicates that it still can lead to a reasonable performance in the real application system.

6.6 Summary

In this chapter, we present an event summarization application for sports games using Twitter streams, integrating the techniques we developed in Chapter 3-5. To make the system applicable in real data, we propose the online version of participant based temporal-content mixture model to conduct sub-event detection. Experiments show that it can achieve similar performance with its batch counterpart.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

This dissertation develops text analysis tools using data mining and machine learning techniques for critical problems in social media. New algorithms are proposed for different problems to address characteristics of text on social media. For each explored problem, related work are reviewed and comprehensive experiments on real datasets and applications are conducted. This dissertation mainly addressed challenges of text analytics on social media as follows:

- Although social media is rich in sentiment text, it is challenging to adapt traditional sentiment analysis techniques, which are conducted on review text, to social media text, because of lack of training data. Active learning can help to reduce the labeling cost. For text data, labels of both documents and words can be utilized to minimize the labeling effort.
- Event detection is critical for text analysis of social media streams to capture the event-related information on social media. Existing methods reply on the volume change of the stream to detect bursts or spikes. However for the social media data, which often contains a lot of noise, these methods are not robust. Combining the information of volume change and topic change of the stream leads to more robust detection results.
- Summarization is an important tools to address information overload problem with a large volume of social media data. In reality, there are various information needs from social media, like comparing two document sets and finding their differences. A versatile summarization model, or a summarization model which can be cus-

tomized, can meet the requirement for a summarizer to generate different summaries for a set of textual posts from different aspects.

Specifically, the following key issues are addressed in this dissertation: (1) utilizing labels of both documents and words to training a classification model with minimized labeling efforts (2) detecting events on data streams of social media, combined the temporal feature, that an event attracts an increasing volume for a short time, and content features, that an event should be a coherent topic (3) summarzing social media posts for different information needs with a versatile summarization framework and a learning-based framework, and (4) building a real-time event summarization and analysis system to utilize text analysis methods in a real application scenario using social media data.

In summary, this dissertation demonstrates and advances the capability of text analysis techniques for various problems on social media. The developed algorithms broadly rely on text classification, ranking, and text clustering and modeling, and they are shown to be effective to be integrated in an real-time social media application.

7.2 Vision for the Future

Social media data plays a more and more important role in our daily lives and in many real applications (e.g., entertainment, health care, disaster management, and scientific discovery). It increases the explosion of information, results in huge amounts of noisy, unstructured, linked, temporal document data on the Internet, and imposes great challenges on text analytics.

My long-term research goal is to continue providing infrastructure of text analytics to help users better understand the large social media data, and enable more developers to build up applications utilizing social media. And in the near future, we will focus on the

following novel problems related to social media, all of which will be built on the thesis work.

- Natural language processing and its evaluation. Natural language processing provides the fundamental basis for the upper layer of text analysis. There are still many classical problems, like co-reference resolution and dis-ambiguity, not yet addressed on the social media data yet. Moreover, although many tools exist, we are lack of the evaluation of them on social media data, so that it is unclear whether they can be applied on the new data with reasonable performance.
- Integration of social network information. Traditional text analysis tasks are usually based on the content of documents. In social media, documents contain not only content but also users information, which further composes the whole social network, so text analysis can base on user profiles and user communities etc. In addition, other typical information of social networks like geotags, and document organization structure like dialogs can be utilized to understand documents more concretely.
- More Applications. Social media has a large impact in a wide range of applications including advertising, disaster management and identification recognization. I believe that these are only a few of the opportunities that a series of better tools of text analytics on social media can provide. I will seek collaborations on various application domains to support the software development of applications based on analysis of social media data.

BIBLIOGRAPHY

[ABHH08] T. Ahlqvist, A. Beck, M. Halonen, and S. Heinonen. Social media roadmaps: Exploring the futures triggered by social media. *VTT Tiedotteita - Research Notes*, (2454), 2008.

[All02] James Allan. Topic detection and tracking: Event-based information organization. *Kluwer Academic Publishers Norwell, MA, USA*, 2002.

[AMP10] J. Attenberg, P. Melville, and F. Provost. A unified approach to active dual supervision for labeling features and examples. *Machine Learning and Knowledge Discovery in Databases*, pages 40–55, 2010.

[APL98] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45. ACM, 1998.

[AW12] S. Aral and D. Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.

[Bal05] Jason Baldridge. The opennlp project, 2005.

[BB98] A. Bagga and B. Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, 1998.

[BJN+02] A.L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3):590–614, 2002.

[BNG11] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 438–441, 2011.

[BNJ03] David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.

[BSR+05] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent.

In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[CAL94] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[CHBG10] M. Cha, H. Haddadi, F. Benevenuto, and P Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 10–17, 2010.

[Chv79] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[CL02] Y.P. Chen and A.L. Liestman. Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks. In *Proceedings of International Symposium on Mobile Ad hoc Networking & Computing*. ACM, 2002.

[CNN+10] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of SIGCHI*, pages 1185–1194, 2010.

[CP11] D. Chakrabarti and K. Punera. Event summarization using tweets. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 66–73, 2011.

[CQL+07] Z. Cao, T. Qin, T.Y. Liu, M.F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[Cun02] Hamish Cunningham. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.

[CWML13] Y. Chang, X. Wang, Q. Mei, and Y. Liu. Towards twitter context summarization with user influence models. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 527–536, 2013.

[CXL+06] Y. Cao, J. Xu, T.Y. Liu, H. Li, Y. Huang, and H.W. Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international*

ACM SIGIR conference on Research and development in information retrieval, pages 186–193. ACM, 2006.

[Dan07] H.T. Dang. Overview of DUC 2007. In *Proceedings of Document Understanding Conference*, pages 1–10, 2007.

[Dhi01] I.S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[DIM06] H. Daumé III and D. Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics, 2006.

[DJZL12] Q. Diao, J. Jiang, F. Zhu, and E.P. Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 536–544, 2012.

[DLPP06] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[DMM08] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602. ACM, 2008.

[DO08] Hoa Trang Dang and Karolina Owczarzak. Overview of the tac 2008 update summarization task. In *Proceedings of Text Analysis Conference*, 2008.

[DSM09] G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 81–90. Association for Computational Linguistics, 2009.

[DTR10] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23th International Conference on Computational Linguistics*, pages 241–249, 2010.

[DWT+14] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 49–54, 2014.

[ER04] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *JAIR*, 22(1):457–479, 2004.

[FCW+11] Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Stephen Hogan, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. #hardtoparse: POS tagging and parsing the twitterverse. In *Proceedings of the AAAI Workshop on Analyzing Microtext*, pages 20–25, 2011.

[Fei98] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[FGM05] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[FISS03] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933–969, 2003.

[Gam04] Michael Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, pages 834–841, 2004.

[GBH09] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[GGLNT04] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, pages 491–501, 2004.

[GHSC04] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. Document classification through interactive supervision of document and term labels. *Knowledge Discovery in Databases: PKDD 2004*, pages 185–196, 2004.

[GK98] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[GMCK00] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 40–48. Association for Computational Linguistics, 2000.

[GN02] M. Girvan and M. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[GSO+11] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47, 2011.

[HGO99] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Neural Information Processing Systems*, pages 115–132, 1999.

[HIMM02] T. Hirao, H. Isozaki, E. Maeda, and Y. Matsumoto. Extracting important sentences with support vector machines. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7. Association for Computational Linguistics, 2002.

[HJ07] B. Han and W. Jia. Clustering wireless ad hoc networks with weakly connected dominating set. *Journal of Parallel and Distributed Computing*, 67(6):727–737, 2007.

[Hof99] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[HTTL13] X. Hu, L. Tang, J. Tang, and H. Liu. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 537–546. ACM, 2013.

[HV09] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technolo-*

gies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 362–370. Association for Computational Linguistics, 2009.

[JM08] D. Jurafsky and J.H. Martin. *Speech and language processing*. Prentice Hall New York, 2008.

[Joa02] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.

[Joa06] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

[Joh73] D.S. Johnson. Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 38–49. ACM New York, NY, USA, 1973.

[JWL+06] F. Jiao, S. Wang, C.H. Lee, R. Greiner, and D. Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 209–216. Association for Computational Linguistics, 2006.

[JYZ+11] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 151–160, 2011.

[KA04] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304. ACM, 2004.

[Kan92] V. Kann. *On the approximability of NP-complete optimization problems*. PhD thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm., 1992.

[KH09] A. M Kaplan and M. Haenlein. The fairyland of second life: Virtual social worlds and how to use them. *Business horizons*, 52(6):563–572, 2009.

[KKT03] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[Kle00] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170. ACM, 2000.

[KM02] K. Knight and D. Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002.

[KPC95] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM, 1995.

[KW06] G. Kossinets and D. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[L.99] Lillian L. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, 1999.

[LCR01] D. Lawrie, W.B. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357, 2001.

[LH03] C.Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78, 2003.

[LHZL09] C. Long, M. Huang, X. Zhu, and M. Li. Multi-document summarization by information distance. In *2009 Ninth IEEE International Conference on Data Mining*, pages 866–871. IEEE, 2009.

[Lin04] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, 2004.

[Liu09] T.Y. Liu. *Learning to rank for information retrieval*. Now Pub, 2009.

[LLM10] J. Leskovec, K. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640, 2010.

[LMP01] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.

[LNK07] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[LZS09] T. Li, Y. Zhang, and V. Sindhwani. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 244–252. Association for Computational Linguistics, 2009.

[Man01] I. Mani. Automatic summarization. *Computational Linguistics*, 28(2), 2001.

[MBB+11] A. Marcus, M. Bernstein, O. Badar, D. Karger, S. Madden, and R. Miller. Twitinfo: Aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 227–236, 2011.

[MC04] T. Mullen and N. Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 412–418, 2004.

[MGL09] P. Melville, W. Gryc, and R.D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1275–1284. ACM, 2009.

[ML03] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced

lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics, 2003.

[MMS93] M. Marcus, M. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[MN98] A.K. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *Machine Learning: Proceedings of the Fifteenth International Conference, ICML*. Citeseer, 1998.

[MS09] P. Melville and V. Sindhwani. Active dual supervision: Reducing the cost of annotating examples and features. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 49–57. Association for Computational Linguistics, 2009.

[MSTPM05] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. An expected utility approach to active feature-value acquisition. In *Data Mining, Fifth IEEE International Conference on*, pages 745–748. IEEE, 2005.

[Nas08] V. Nastase. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 763–772. Association for Computational Linguistics, 2008.

[NMD12] J. Nichols, J. Mahmud, and C. Drews. Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM Interntional Conference on Intelligent User Interfaces*, pages 189–198, 2012.

[NV05] A. Nenkova and L. Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*, 2005.

[NVM06] A. Nenkova, L. Vanderwende, and K. McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580. ACM, 2006.

[OER05] J. Otterbacher, G. Erkan, and D.R. Radev. Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on*

Human Language Technology and Empirical Methods in Natural Language Processing, pages 915–922. Association for Computational Linguistics, 2005.

[OKA10] B. O'Connor, M. Krieger, and D. Ahn. TweetMotif: Exploratory search and topic summarization for twitter. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 384–385, 2010.

[OLL07] Y. Ouyang, S. Li, and W. Li. Developing learning strategies for topic-based summarization. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 79–86. ACM, 2007.

[OOD+13] O. Owoputi, B. OConnor, C. Dyer, K. Gimpel, N. Schneider, and N. Smith. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*, pages 380–390, 2013.

[PLV02] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the conference on Empirical methods in natural language processing*, pages 79–86. Association for Computational Linguistics, 2002.

[POL10] S. Petrovic, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189, 2010.

[PRV07] P. Pingali, K. Rahul, and V. Varma. IIIT Hyderabad at DUC 2007. In *Proceedings of DUC 2007*, 2007.

[RA07] H. Raghavan and J. Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 79–86. ACM, 2007.

[RCME11] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, 2011.

[Reh13] Ines Rehbein. Fine-grained pos tagging of german tweets. In *Language Processing and Knowledge in the Web*, pages 162–175. Springer, 2013.

[RJST04] D.R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938, 2004.

[RMEC12] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1104–1112, 2012.

[RMJ06] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on features and instances. *The Journal of Machine Learning Research*, 7:1655–1686, 2006.

[RS97] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 475–484. ACM New York, NY, USA, 1997.

[SBC03] H. Saggion, K. Bontcheva, and H. Cunningham. Robust generic and query-based summarisation. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 235–238, 2003.

[Set09] B. Settles. Active Learning Literature Survey. *Technical Report 1648*, 2009.

[SHM09] V. Sindhwani, J. Hu, and A. Mojsilovic. Regularized co-clustering with dual supervision. In *Advances in Neural Information Processing Systems*, pages 1505–1512, 2009.

[SL10] C. Shen and T. Li. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 984–992. Association for Computational Linguistics, 2010.

[SL11a] C. Shen and T. Li. Learning to rank for query-focused multi-document summarization. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 626–634. IEEE, 2011.

[SL11b] C. Shen and T. Li. A non-negative matrix factorization based approach for active dual supervision from document and word labels. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 949–958. Association for Computational Linguistics, 2011.

[SLWL13] C. Shen, F. Liu, F. Weng, and T. Li. A participant-based approach for event summarization using twitter streams. In *Proceedings of NAACL-HLT*, pages 1152–1162, 2013.

[SM08] V. Sindhwani and P. Melville. Document-word co-regularization for semi-supervised sentiment analysis. In *Proceedings of Data Mining, Eighth IEEE International Conference on*, pages 1025–1030. IEEE, 2008.

[SML09] V. Sindhwani, P. Melville, and R.D. Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 953–960. ACM, 2009.

[SMR07] C. Sutton, A. McCallum, and K. Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *The Journal of Machine Learning Research*, 8:693–723, 2007.

[SP03] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics, 2003.

[SSL+07] D. Shen, J.T. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *Proceedings of IJCAI*, volume 7, pages 2862–2867, 2007.

[STUB08] T. Sandler, P.P. Talukdar, L.H. Ungar, and J. Blitzer. Regularized learning with networks of features. *Advances in Neural Information Processing Systems*, pages 1401–1408, 2008.

[TGRM08] M. Taylor, J. Guiver, S. Robertson, and T. Minka. SoftRank: optimizing non-smooth rank metrics. In *Proceedings of the international conference on Web search and web data mining*, pages 77–86. ACM, 2008.

[TK02] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[TKMS03] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram 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*, pages 173–180. Association for Computational Linguistics, 2003.

[TLT+11] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM, 2011.

[TSWY09] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816, 2009.

[TYC09] J. Tang, L. Yao, and D. Chen. Multi-topic based Query-oriented Summarization. In *Proceedings of SDM*, pages 1147–1158, 2009.

[TYO11] Hiroya Takamura, Hikaru Yokono, and Manabu Okumura. Summarizing a document stream. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, pages 177–188, 2011.

[TZTX07] M.T. Thai, N. Zhang, R. Tiwari, and X. Xu. On approximation algorithms of k-connected m-dominating sets in disk graphs. *Theoretical Computer Science*, 385(1-3):49–59, 2007.

[Wan09] Xiaojun Wan. Topic analysis for topic-focused multi-document summarization. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1609–1612. ACM, 2009.

[WL01] J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems*, 18(1):13–36, 2001.

[WL11] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 401–408, 2011.

[WLJH10] J. Weng, E.P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270, 2010.

[WLLH08] F. Wei, W. Li, Q. Lu, and Y. He. Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 283–290. ACM, 2008.

[WLZD08] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314. ACM, 2008.

[WWH05] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354, 2005.

[WX09] X. Wan and J. Xiao. Graph-Based Multi-Modality Learning for Topic-Focused Multi-Document Summarization. In *Proceedings of IJCAI*, pages 1586–1591, 2009.

[WYX07a] X. Wan, J. Yang, and J. Xiao. Manifold-ranking based topic-focused multi-document summarization. In *Proceedings of IJCAI*, pages 2903–2908, 2007.

[WYX07b] X. Wan, J. Yang, and J. Xiao. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics*, pages 543–552, 2007.

[WZLG09a] D. Wang, S. Zhu, T. Li, and Y. Gong. Comparative document summarization via discriminative sentence selection. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1963–1966. ACM, 2009.

[WZLG09b] D. Wang, S. Zhu, T. Li, and Y. Gong. Multi-document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 297–300, 2009.

[XCLH06] J. Xu, Y. Cao, H. Li, and Y. Huang. Cost-sensitive learning of SVM for ranking. In *Proceedings of ECML*, pages 833–840, 2006.

[YC10] Jiang Yang and Scott Counts. Predicting the speed, scale, and range of information diffusion in twitter. *ICWSM*, pages 355–358, 2010.

[YL10] Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *Data Mining, 2010 IEEE 10th International Conference on*, pages 599–608. IEEE, 2010.

[YPC98] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36. ACM, 1998.

[ZE08] Omar F. Zaidan and Jason Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 31–40, 2008.

[ZGD+11] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu. Combining lexicon-based and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89, 2011.

[ZH03] L. Zhou and E. Hovy. A web-trained extraction summarization system. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 205–211. Association for Computational Linguistics, 2003.

[ZHD+01] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 25–32. ACM, 2001.

[ZHW05] L. Zhao, X. Huang, and L. Wu. Fudan University at DUC 2005. In *Proceedings of DUC*, 2005.

[ZSAG12] Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. Towards real-time summarization of scheduled events from twitter streams. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, pages 319–320, 2012.

[ZZW+12] Siqi Zhao, Lin Zhong, Jehan Wickramasuriya, Venu Vasudevan, Robert LiKamWa, and Ahmad Rahmati. Sportsense: Real-time detection of NFL game events from twitter. *Technical Report TR0511-2012*, 2012.

[ZZWV11] Siqi Zhao, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. Human as real-time sensors of social and physical events: A case study of twitter and sports games. *Technical Report TR0620-2011, Rice University and Motorola Labs*, 2011.

VITA

CHAO SHEN

2006 B.S. of Computer Science
Fudan University
Shanghai, P.R.China

2009 M.S. of Computer Application Technology
Fudan University
Shanghai, P.R.China

2009-2014 Doctoral Candidate
Florida International University
Miami, FL, USA

PUBLICATIONS

- Wubai Zhou, Chao Shen, Tao Li, Shu-Ching Chen, Ning Xie, Jinpeng Wei. Generating textual storyline to improve situation awareness in disaster management. In *Proceedings of 2014 IEEE 13th International Conference on Information Reuse and Integration*, 2014

- Wubai Zhou, Chao Shen, Tao Li, Shu-Ching Chen, Ning Xie, Jinpeng Wei. A Bipartite-Graph Based Approach for Disaster Susceptibility Comparisons among Cities. In *Proceedings of 2014 IEEE 13th International Conference on Information Reuse and Integration*, 2014

- Li Zheng, Chao Shen, Liang Tang, Chunqiu Zeng, Tao Li, Steve Luis, and Shu-Ching Chen. Data Mining Meets the Needs of Disaster Information Management. In *IEEE Transactions on Human-Machine Systems on 43 (5)*, 451-464, 2013

- Chunqiu Zeng, Yexi Jiang, Li Zheng, Jingxuan Li, Lei Li, Hongtai Li, Chao Shen, Wubai Zhou, Tao Li, Bing Duan, Ming Lei, and Pengnian Wang. FIU-Miner: A Fast, Integrated, and User-Friendly System for Data Mining in Distributed Environment. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1506-1509, 2013

- Chao Shen, Fei Liu, Fuliang Weng and Tao Li. A Participant-based Approach for Event Summarization Using Twitter Streams. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152-1162, 2013

- Li Zheng, Chao Shen, Liang Tang, Chunqiu Zeng, Tao Li, Steve Luis, Shu-Ching Chen and Jainendra K. Navlakha. Disaster SitRep - A Vertical Search Engine and Information Analysis Tool in Disaster Management Domain. In *Proceedings of 2012 IEEE 13th International Conference on Information Reuse and Integration*, pages 457-465, 2012

- Chao Shen, and Tao Li. Learning to Rank for Query-focused Multi-document Summarization. In *Proceedings of 2011 IEEE 11th International Conference on Data Mining*, pages 626-634, 2011

- Chao Shen, and Tao Li. A Non-negative Matrix Factorization Based Approach for Active Dual Supervision from Document and Word Labels. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011.

- Chao Shen, Tao Li, and Chris H.Q. Ding. Integrating Clustering and Multi-Document Summarization by Bi-mixture Probabilistic Latent Semantic Analysis (PLSA) with Sentence Bases. In *Proceedings of the 25th AAAI conference on artificial intelligence*, pages 914-920, 2011.

- Li Zheng, Chao Shen, Liang Tang, Tao Li, Steve Luis, and Shu-Ching Chen. Applying data mining techniques to address disaster information management challenges on mobile devices. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 283-291, 2011.

- Chao Shen, Dingding Wang, and Tao Li. Topic Aspect Analysis for Multi-Document Summarization. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1545-1548, 2010.

- Chao Shen and Tao Li, Multi-Document Summarization via the Minimum Dominating Set. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 984-992, 2010.

- Li Zheng, Chao Shen, Liang Tang, Tao Li, Steve Luis, Shu-Ching Chen, and Vagelis Hristidis. Using Data Mining Techniques to Address Critical Information Exchange Needs in Disaster Aected Public-Private Networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010.

- Lei Li, Dingding Wang, Chao Shen, and Tao Li. Ontology-Enriched Multi-document Summarization in Disaster Management. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 819-820, 2010.