Topical analysis of text streams

He, Qi

2009

He, Q. (2009). Topical analysis of text streams. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/17764>

<https://doi.org/10.32657/10356/17764>

NANYANG TECHNOLOGICAL UNIVERSITY

Topical Analysis of Text Streams

HE, Qi

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of

Doctor of Philosophy

June 2008

Abstract

Topic detection (TD) is an important area of research whose primary goal is to detect retrospective or new topics from a stream of news articles. It could be extremely useful in many applications including news aggregation portals, news alert systems, event search engine, terrorist activity tracking, etc. However, specialists who analyze news articles have a hard time separating the wheat from the chaff, due to the overwhelming amount of news streams (over 10,000 as of 2008). For many years, Topic Detection has been tackled as a clustering task by the TDT (Topic Detection and Tracking) research community. However, time, which plays a pivotal role in news articles has never been given due consideration in the past. In this research we present a thorough study on various temporal topic detection models that explicitly incorporate the element of time. We further discovered that bursty temporal word features play an important role in improving topic detection performance, and ventured to provide an in-depth analysis and systematic categorization of all word features into 5 general types using techniques from signal processing. Armed with a small set of extracted bursty features from historical or online news streams, we proposed a number of effective algorithms to detect topics from a news stream in both offline and online modes. Our algorithms are mathematically elegant, simple, and extremely practical, when benchmarked against some of the best topic detection models including spherical k -means, Latent Dirichlet Allocation (LDA), and von-Mises Fisher mixtures. Finally, we present a case study of a personalized news alert application, where subscribers can specify interesting anticipatory events, and show how a simple supervised event transition classifier can be used to effectively identify user anticipated events. Our research is one of the most comprehensive studies on both offline and online topic detection, of which the latter has been an open research problem for many years. In fact, our online topic detection model can be viewed as a significant advancement in the field, which paves the way for further improvements by other TDT experts.

Acknowledgments

I would like to take this opportunity to express my appreciation to the following people.

Dr Kuiyu Chang, my supervisor, for his patience, guidance, friendship, generosity, and support for the past three and half a years. It was a great pleasure and yet humbling experience to work under Dr Chang, and I learnt a great deal from observing his philosophy and way of handling both technical and non-technical aspects of life.

Dr Ee-Peng Lim, my co-advisor, for the weekly seminars where we discussed research ideas. I especially appreciate his meticulous review of my work and his invaluable research insight. His strict supervision also encouraged me to keep on pursuing my Ph.D. study with vigor.

Dr Jun Zhang, my best friend, for the confidence that he helped instilled in me from his frequent encouragements and suggestions.

Dr Arindam Banerjee who spared his precious time to discuss with me about my ideas and gave me useful suggestions.

I would like to thank **My Family** including my mother and my sister for their constant love, encouragement, and support. Special appreciation is due to my father, **Mr. Caiming He** who is a true gentleman, for constantly challenging me to continue my Ph.D. study, which served as a much needed gravitational force for putting me back on track during the numerous times that I got distracted. Thanks are also due to my wife, **Ms. Linjun Chen** for her constant encouragement, love, support, companionship, and care during this long journey, especially during the last few years of my Ph.D. study.

Last but not least, I am grateful to many of my friends who spared time to share their thoughts and joy with me. Also, I would like to thank **Nanyang Technological University**, the **School of Computer Engineering**, and **Microsoft Research Asia** for providing financial support and world-class research facilities.

Contents

Abstract	i
Acknowledgments	ii
List of Figures	vii
List of Tables	xi
List of Algorithms	xiii
List of Notations	xiv
List of Abbreviations	xvi
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problem Definition	4
1.3 Challenges and Our Contributions	6
1.4 Thesis Organization	9
2 Related Work	11
2.1 Background of TDT Community	11
2.1.1 History of TDT	11
2.1.2 Document Representation in TDT	13
2.1.3 Similarity Measurement in TDT	14
2.1.4 Topic Detection in TDT and Clustering	15
2.1.5 Novelty of Our Research	17
2.2 Probabilistic Models for Topic Detection	17
2.2.1 Naive Bayes Model	17
2.2.2 Generative Probabilistic Models	18
2.2.3 Novelty of Our Research	19

2.3	Word-level Topic Detection	20
2.3.1	TDT Approaches Utilizing Bursts	20
2.3.2	Trajectory Analysis of Bursty Phenomena	21
2.3.3	Novelty of Our Research	22
2.4	Intra-topic Analysis	22
2.4.1	Event Transition Graph (ETG)	23
2.4.2	Domain Knowledge Based Topic Detection	24
2.4.3	Personalized TDT	24
2.4.4	Novelty of Our Research	25
2.5	Other Topic Detection Research	25
2.6	Summary	26
3	Discriminative Features of News Streams and Word-Level Topic Detection	27
3.1	Motivations of Discriminative Features	27
3.2	Problem Definition for Word-Level Topic Detection	30
3.3	Modeling Feature Trajectory	31
3.3.1	Feature Trajectory Vector	31
3.3.2	Bursty Features	32
3.3.3	Spectral Analysis on Feature Trajectory	33
3.4	Identifying Bursty Features from a Historical Stream	36
3.4.1	Categorizing Features based on Power Spectrum and Periodicity . .	36
3.4.2	Detecting Bursts via Density Models	41
3.5	Reconstructing Offline Topics	42
3.5.1	Correlations Among Discriminative Bursty Features	43
3.5.2	Unsupervised Greedy Topic Detection	44
3.6	From Online Bursty Features to Online Topics	44
3.6.1	Dominant Period Trend Analysis for Online Bursty Features	45
3.6.2	Modeling The Dominant Period Trend with a 3-state Automata . .	49
3.6.3	Maximum Likelihood Bursty Feature Detection	52
3.6.4	Predicting Online Topics	56
3.7	Performance Evaluation	56

3.7.1	Dataset and Experimental Setup	56
3.7.2	Retrospectively Categorizing Features	57
3.7.3	Retrospectively Detecting Aperiodic Topics	59
3.7.4	Retrospectively Detecting Periodic Topics	61
3.7.5	Detecting Online Bursty Features	63
3.7.6	Superiority of the Online Bursty Feature Detection Method	63
3.7.7	Detecting Online Aperiodic Topics	65
3.7.8	Detecting Online Periodic Topics	68
3.8	Summary	71
4	Temporal Discriminative Probabilistic Models for Document-Level Topic Detection	73
4.1	Motivation for Temporal Discriminative Probabilistic Models	73
4.2	Document-Level Topic Detection Model	75
4.3	Topic Detection Models	77
4.3.1	Non-probabilistic Models	77
4.3.2	Mixture Generative Models	79
4.3.3	Mixed Membership Generative Models	82
4.4	Discriminative Probabilistic Models	84
4.4.1	Offline Discriminative Model	85
4.4.2	Online Discriminative Model	87
4.4.3	Explore Discriminative Features	91
4.5	Performance Evaluation	95
4.5.1	Dataset and Data Preprocessing	95
4.5.2	Experiment 1: Offline Topic Detection - Hard Partition	96
4.5.3	Experiment 2: Offline Topic Detection - Soft Partition	106
4.5.4	Experiment 3: Online Topic Detection	110
4.6	Discussions	116
4.7	Summary	117

5	Intra-topic Transition Detection via Classification	119
5.1	Motivations and Problem Definition	119
5.2	The AED Framework	120
5.2.1	AED on Document Streams	120
5.2.2	Types of Anticipatory Event Detection	121
5.2.3	AED Prototype	122
5.3	Modeling the AET	124
5.3.1	Sentence Classifier	124
5.3.2	Document Classifier	126
5.4	Dataset	128
5.4.1	<i>Basket100</i> Dataset	128
5.4.2	<i>Google Acquisition</i> Dataset	130
5.4.3	<i>Acquisition7</i> dataset	131
5.4.4	<i>CNPC_Acquires_PKZ</i> dataset	132
5.5	Exploratory Data Analysis	134
5.5.1	FDA and KFDA Visualization Analysis	134
5.5.2	Analysis of Acquisition Dataset	143
5.6	Experiments on AED	150
5.6.1	Experimental Setup	150
5.6.2	Sentence AED Model	151
5.6.3	Experiments on Document AED Model	155
5.7	Summary	161
6	Conclusion	166
6.1	Discussion	166
6.2	Future Directions	168
6.2.1	Discriminative Word Feature Selection	168
6.2.2	Probabilistic Document-Level Models	169
6.2.3	Anticipatory Event Detection	169
A	List of Publications	170
	References	172

List of Figures

1.1	Topical detection on news stream.	4
1.2	Different topic (user preference) resolutions for an “earthquake” topic. . . .	5
3.1	Feature correlation (Normalized DF versus time) between a) <i>Easter</i> and <i>April</i> b) <i>Unaudited</i> and <i>Ended</i>	28
3.2	Feature <i>Christmas</i> with relative high S_x and long-term P_x	37
3.3	Feature <i>soccer</i> with relative high S_x and short-term P_x	37
3.4	Feature <i>DBS</i> with relative low S_x and long-term P_x	38
3.5	Feature <i>how</i> as an example confusing with feature “soccer”.	39
3.6	Feature <i>beenb</i> with relative low S_x and short-term P_x	39
3.7	The 5 feature sets for topics.	40
3.8	Modeling <i>Iraq</i> ’s time series as a truncated Gaussian with $\mu = 09/06/1996$ and $\sigma = 6.26$	41
3.9	HH example for the trend of P_x	46
3.10	HL example for the trend of P_x	46
3.11	LH example for the trend of P_x	47
3.12	LL example for the trend of P_x	47
3.13	SW example for 2-state of P_x	47
3.14	The dominant period trend of the word <i>Unaudited</i>	48
3.15	3-state automaton for capturing the dominant period trend.	50
3.16	Effect of sliding window on the <i>stable</i> state.	52
3.17	Effect of a sliding window of size $l = 7$ on other features.	53
3.18	Effect of the DFIDF trajectory gradient on the <i>stable</i> state.	54
3.19	Largest change in DFIDF trajectory gradient helps to avoid local maxima. .	54
3.20	Distribution of SW (stopwords) in the HH, HL, LH, and LL regions.	57

3.21	Distribution of categorized features over the four quadrants (shading in log scale).	58
3.22	Distribution of online bursty features (# detected bursty features:day). . .	64
3.23	Categorization of detected online bursty features.	64
3.24	Distribution of online aperiodic topics (# detected topics:day).	66
3.25	Cost categorization of detected online aperiodic topics.	66
3.26	Online aperiodic topics with minimal cost per 30 days (topic cost:day). . . .	68
3.27	Distribution of online periodic topics (# detected topics:day).	69
3.28	Categorizations for detected online periodic topics.	69
3.29	Online detection status versus offline ground truth for c_{23} . Bars denote periods. Delayed alarm.	70
3.30	Refined online detection status versus historical ground truth for c_{24} . Bars denote periods. False alarm in the first period and delayed alarm starting from the second period.	70
3.31	Online detection status versus offline ground truth for c_{25} . Bars denote periods. Delayed alarm.	70
3.32	Online detection status versus offline ground truth for c_{26} and c_{27} . Bars denote periods. Delayed alarm.	71
4.1	Explicit dependencies among document (d), descriptor (x), and time (t). . .	89
4.2	Document frequency evolutions for three words. The x-axis shows the publication date, and the y-axis gives the normalized document frequency at each day. $N_x(t)$ is the number of documents containing word x at time t , and $N(t)$ is the number of documents at time t	92
4.3	The burstiness of common, bursty, and rare words in the TDT3 dataset. The x-axis is the date, and the y-axis is the variation of average document frequency score over all words in the same category.	93
4.4	Topic distribution of documents in TDT3-Single	95
4.5	The cumulative count of topics versus day for TDT3-Single	96
4.6	Offline topic detection time complexity versus number of clusters k (log-scaled). .	100
4.7	Internal Similarities across various k on offline topic detection.	101
4.8	Gradient of Internal Similarity across various k on offline topic detection. .	101

4.9	Perplexity gradient across various k for LDA offline topic detection.	102
4.10	Cluster purity across various k on offline topic detection.	103
4.11	Cluster entropy across various k on offline topic detection.	104
4.12	Class purity across various k on offline topic detection.	104
4.13	Class entropy across various k on offline topic detection.	105
4.14	nMI across various k on offline topic detection.	105
4.15	Precision and recall across various top m soft clusters.	108
4.16	F-measure across various top m soft clusters.	109
4.17	Online topic detection time complexity (log-scaled).	113
4.18	The DET curve for online topic detection.	114
5.1	AED for document streams.	121
5.2	An Event Transition Graph (ETG) for the “acquisition” topic genre.	121
5.3	The four AED scenarios.	123
5.4	Online AED system showing only the selected AET of the ETG.	123
5.5	2-level SVM sentence classifier.	125
5.6	The ETG for the <i>Basket100</i> dataset.	128
5.7	The ETG for the <i>Google Acquisition</i> dataset.	130
5.8	FDA for document-level dataset	137
5.9	FDA for sentence-level dataset (class 1 vs. class 2)	138
5.10	FDA for sentence-level dataset (class 1 vs. class 3)	139
5.11	FDA for sentence-level dataset (class 2 vs. class 3)	140
5.12	KFDA (cosine similarity matrix) for document-level dataset	141
5.13	KFDA (cosine similarity matrix) for sentence-level dataset (class 1 vs. class 2)	143
5.14	10-fold cross-validation results on the <i>CNPC_Acquires_PKZ</i> dataset.	144
5.15	1-level SVM classification with 10-fold cross-validation.	152
5.16	Results for the first level of the 2-level SVM classifier with 10-fold cross-validation.	153
5.17	Results for the second level of the 2-level SVM classifier with 10-fold cross-validation.	153
5.18	Overall results of 2-level SVM classifiers with 10-fold cross-validation.	154
5.19	Comparison of F1-Measures between 1-level and 2-level classification	155

5.20	AED using cosine similarity on Acquisition7 Dataset.	157
5.21	Average test results of the four text classification methods applied to <i>Google Acquisition</i>	158
5.22	$t_{2,3}$ boundaries in <i>Acquisition7</i> dataset.	160
5.23	Online AED evaluation on acquisition event: Adobe acquire Macromedia. $t(d_m) = t(e_3)$, an accurate alarm.	161
5.24	Online AED evaluation on acquisition event: CNPC acquires PKZ. $t(d_m) =$ $t(e_3)$, an accurate alarm.	161
5.25	Online AED evaluation on acquisition event: eBay acquires Skype. $t(d_m) =$ $t(e_3)$, an accurate alarm.	162
5.26	Online AED evaluation on acquisition event: Lenovo acquires IBM PC Division. $t(d_m) < t(e_3)$, a false alarm.	162
5.27	Online AED evaluation on acquisition event: Oracle acquires PeopleSoft. $t(d_m) < t(e_3)$, a false alarm.	163
5.28	Online AED evaluation on acquisition event: Oracle acquires Siebel. $t(d_m) =$ $t(e_3)$, an accurate alarm.	163
5.29	Online AED evaluation on acquisition event: SBC acquires AT&T. $t(d_m) >$ $t(e_3)$, a delayed alarm.	164

List of Tables

3.1	All important aperiodic topics ($c_1 - c_{17}$).	60
3.2	Top 5 less-reported aperiodic topics ($c_{18} - c_{22}$).	61
3.3	Top 5 important periodic topics ($c_{23} - c_{27}$).	62
3.4	Online topic detection benchmarks.	65
3.5	The online aperiodic topics ($c_{28} - c_{39}$) with minimal cost per 30 days. . . .	67
4.1	Probabilistic TD models.	75
4.2	Comparisons between offline/batch and online topic models.	77
4.3	Bursty score examples.	94
4.4	Distribution of documents spanning two or more topics TDT3-Multiple . .	96
4.5	A sample contingency table (2×3) for topic detection.	99
4.6	Contingency table for document pairs.	107
4.7	Minimal DET cost values of all models for online topic detection.	115
4.8	False alarms under <i>zero</i> miss rate of all models for offline topic detection. .	115
4.9	Model complexity.	117
5.1	Class distribution of <i>Basket100</i> dataset.	129
5.2	Sparsity statistics for <i>Basket100</i> dataset.	130
5.3	The make up of <i>Acquisition7</i> dataset.	132
5.4	10-fold cross-validation results for FDA on document-level dataset.	138
5.5	10-fold cross-validation results for FDA on sentence-level dataset (class 1 vs. class 2)	139
5.6	10-fold cross-validation results for FDA on sentence-level dataset (class 1 vs. class 3)	139
5.7	10-fold cross-validation results for FDA on sentence-level dataset (class 2 vs. class 3)	140

5.8	10-fold cross-validation results for KFDA (cosine similarity matrix) on document-level dataset	142
5.9	10-fold cross-validation results for KFDA (cosine similarity matrix) on sentence-level dataset (class 1 vs. class 2)	142
5.10	Average classification results.	144
5.11	Average evaluation for both positive and negative classes.	145
5.12	Overall summarization of 5 errors in <i>CNPC_Acquires_PKZ</i> dataset.	147
5.13	Event-conditioned novelty detection on <i>Detection of Anticipatory Event Transition</i> for <i>Acquisition7</i> dataset.	157
5.14	Average test results on <i>Google Acquisition</i>	158
5.15	Summary of AED results on detecting $t_{2,3}$ for <i>Acquisition7</i> using the VOT-ING method.	164
5.16	Summary of AED results on detecting $t_{2,3}$ for <i>Acquisition7</i> using the CON-TENT method.	165

List of Algorithms

3.1	<u>H</u> euristic <u>S</u> topwords detection (HS)	40
3.2	<u>U</u> nsupervised <u>G</u> reedy topic detection (UG).	45
4.1	(Offline Topic Detection) (D, ϵ)	98
4.2	(Online Topic Detection) (D, ϵ)	111

List of Notations

$\mathbb{R}^n$	:	n -dimensional space of real numbers
N	:	Number of (training) samples (documents)
n	:	Size of the whole (word) feature corpus
S	:	Number of (training) sentences in a data collection
T	:	Time period spanned by a data collection
t	:	Random time point within T
τ	:	Time $t' \leq t$
k	:	Number of components in a mixture model
H	:	Hypothesis
$\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_N]$	:	Word-by-document matrix of (training) samples (document)
$\mathbf{d}_i = [w_{1i}, \dots, w_{ni}]^T$	:	The i -th sample vector in $\mathbf{D}$
$\mathbf{d} = [w_1, \dots, w_n]^T$	:	Vector of random sample in $\mathbf{D}$
$D = \{d_1, \dots, d_N\}$	:	Set of (training) samples (documents)
d_i	:	The i -th sample in D
d	:	Random sample in D
$F = \{x_1, \dots, x_n\}$	:	The (word) feature space
x_i	:	The i -th feature of F
x	:	Random feature of F
$X \in F$	:	Set of discriminative features
D_i	:	Set of samples containing feature x_i
w_{ji}	:	Weight of element at the j -th row and the i -th column in matrix $\mathbf{D}$
$w_{x,d}$	:	Weight of feature x in document d
$f_{x,d}$	:	Frequency of feature x in document d
$f_{x,s}$	:	Frequency of feature x in sentence s
$f_{x,z}$	:	Overall frequency of feature x for class z

$C = \{c_1, \dots, c_{ C }\}$	: Set of clusters in D
c_i	: The i -th cluster in C
$\mathbf{c}_i$	: The i -th cluster's mean vector in C
c	: Random cluster in C
$Z = \{z_1, \dots, z_{ Z }\}$	: Set of classes in a classification problem; set of classes in D
z_i	: The i -th class in Z
$\mathbf{z}_i$	: The i -th class's mean vector in Z
z	: Random class in Z
$E = \{e_1, \dots, e_k\}$	: A sequence of events for a certain topic genre
e_i	: The i -th event in E
e	: Any event in E
$t_{i,j}$	: Event transition from e_i to e_j in E
N_x	: Number of samples for which the feature x exists in D
$N(c_i)$	: Number of samples clustered into cluster c_i in D
$N(z_i)$	: Number of samples for which the true class is z_i in D
$N_x(c_i)$	: Number of samples for which the feature x exists within cluster c_i in D
$N_x(z_i)$	: Number of samples for which the feature x exists within class z_i in D
$N(t)$	: Number of samples at time t in D
$N(c_i, t)$	: Number of samples clustered into cluster c_i at time t in D
$N(\tau)$	: Number of samples within before time τ in D
$N_x(t)$	: Number of samples for which the feature x exists at time t in D
$N_x(\tau)$	: Number of samples for which the feature x exists before time τ in D
λ	: Clamping factor
μ	: Mean
σ	: Standard deviation
κ	: Concentration parameter of vMF distribution
ϵ	: Tolerance for stopping at a given iteration
R_n	: A (normalizing) constant related to n ; n is optional
$\#$	: Number/count of
$ \cdot $	: Cardinality of a set
$t(\cdot)$	: Publication date/time
A	: Automaton
q	: State in A
$\mathbf{q}$	: State sequence in A
$y_x = [y_x(1), y_x(2), \dots, y_x(T)]$	: Trajectory vector of feature x

List of Abbreviations

AED	:	Anticipatory Event Detection
AET	:	Anticipatory Event Transition
DET	:	Detection Error Tradeoff curve
DF	:	Document frequency
DFT	:	Discrete Fourier Transform
DP	:	Dominant period
DPS	:	Dominant power spectrum
DIS	:	Discriminative probabilistic models
ETG	:	Event Transition Graph
FDA	:	Fisher Linear Discriminant Analysis
IDF	:	Inverse document frequency
ISF	:	Inverse sentence frequency
KFDA	:	Kernel Fisher Linear Discriminant Analysis
LDA	:	Latent Dirichlet Allocation
NE	:	Named entity
SPK	:	Spherical k -means clustering model
SVM	:	Support Vector Machine
TD	:	Topic detection
TDT	:	Topic detection and tracking
TF	:	Term frequency
vMF	:	von-Mises Fisher distribution

Chapter 1

Introduction

1.1 Background and Motivation

Media Bubble: *People Like News, Especially When It's Pretty.*

Media conglomerates like CNN and MSNBC have played an important role in the daily lives of people by reporting the latest happenings from around the globe. Moreover, with close to a quarter of the world's population online, the Internet has become the de facto source of news information. Further, with the advent of mobile Internet-enabled devices such as GPRS, 3G and 4G Internet enabled mobile phones, the Internet has gradually evolved into the next real-time universal information source. For example, there are more than 10,000 official or semi-official online news sources. This huge selection of online news sources has completely altered the way in which people consume news. In fact, professionals from time-critical industries such as the financial sector, who need real-time information for decision-making, are already using SMS (Short Message Service) as a 24-hour news delivery medium. Traditional news providers like BBC and CNN have also migrated to the Internet, providing flashier features like interactive graphics and video. At Yahoo News, for instance, users view 200 million videos and 800 million photos every month.

However, paradigm shifts always come with new challenges. Ubiquitous information access comes with the peril of ubiquitous bombardment of potentially useless information. Imagine that we have to separate wheat from tons of chaff, which is exactly how it feels like to select relevant news from among the vast amount of news articles. Therefore, many

CHAPTER 1. INTRODUCTION

news aggregators, like Yahoo, Google and AOL, have set up news aggregation sites that aggregate headlines from thousands of multi-lingual news sources worldwide. Similar stories are grouped together and displayed according to a user's personal preferences. While Yahoo and AOL rely on human editors to select and package articles, Google News is entirely automated. The trade-offs are obvious, automated methods are up-to-date and have wide coverage but low accuracy, while human editors create more precise grouping but covers significantly fewer topics. If we prefer high recall (low miss rate, i.e., covering every important topic), it would be impossible for human editors to review even a small subset. Thus, human editors can generate accurate news alerts or aggregates with near perfect precision, but could miss out (low recall) on many important events.

What is the best way of organizing news? Should it be the fully manual approach of Yahoo, or the completely automated style of Google news? This is akin to asking if you prefer high precision or high recall. We do not have the exact answers, but one thing we know for sure is this. The practical reason why Yahoo still uses humans is simply because automated methods are unsatisfactory! If we can gradually improve the precision of automated approaches beyond a certain tipping point, it will eventually be acceptable. This is precisely the philosophy behind our research.

One of the most widely-accepted principles might be the concept of a topic. For instance, the building of two world-class casinos in Singapore may give rise to a series of duplicated reports by major national and international media sources like The Strait Times, Zaobao, and Newsweek, etc. Even within the same publication, the topic could be reported over a span of few days to several months or even years, depending on the impact of the topic itself. A cluster of news articles on the same topic has many practical appeals for news readers, summarized as follows.

- Since no user can or will read all of the news published in the world, most people naturally look at news from topics that interest them. For example, hedge fund managers may be interested in any topic related to the casinos in Singapore, while teenagers may only be interested in the latest fashion and gadget trends.
- Articles on the same topic can be gathered from many different sources and over a time period, thereby enabling a user to directly access all articles directly or indirectly related to it. Such a grouping of like themed articles also facilitate search. For example, users can search within the topic to find documents from a specific source.

CHAPTER 1. INTRODUCTION

- Within each identified topic, sub-topics that may develop over time can be tracked. For example, the casino topic went through the following phases: government announces feasibility study of operating a casino in Singapore, government releases result of studies, government solicits public opinion, government debate issue in parliament, parliament votes on issue, cabinet announces final decision, prime minister addresses nation, tender for casinos announced, bidding begins, successful bidders announced, construction of casinos begin, casino opens for business, crime rate increases, etc. These phases can be tracked by an automated system, and alerts sent to users when an anticipatory event occurs. For example, people in the construction sector may be more interested in the casino bidding phase than any others, and may want to know firsthand if a bid is announced. Without a topical grouping of similar documents, further analysis of the evolution of sub-topics would not be possible.
- After grouping news articles into topics, we can offer users as many perspectives on a story as possible. For example, the government's decision to build a casino are strongly supported by mainstream media in Singapore, but a variety of other views may be discovered from non-mainstream news sources, which can be detected from among all documents on the same topic.
- The detected topics can be summarized with just a few representative keywords or sentences. The summaries allow users to quickly glance over a large number of topics without actually reading them. For example, once we detected the casino topic, we could find representative words like IR, casino, two, integrated resort, government, 2009, sentosa, marina, sands, las vegas, etc. Just by looking at topic keywords, a user can immediately decide if a topic interests them. A side effect of topic aggregation is that it effectively steals web traffic away from a single news source, eroding what little control traditional news publishers have over users. The result is a total liberalization of information, where news are no longer monopolized by the mainstream media.

In summary, topical analysis of news articles are vastly useful for personalized news, news alerts, news feeds on a specific sub-topic, and news archive searching and tracking, etc.

1.2 Problem Definition

Good topic detection techniques are essential for the timely and effective dissemination of information from heterogenous news sources. A system that assigns each news article to a topic is therefore very helpful for rejecting relevant information and detecting topics in a specific domain, as shown in Figure 1.1.

Figure 1.1: Topical detection on news stream.

Given a news stream D , there exists an unknown number of topics Z . Our task is to uncover a set of clusters C from D , so that C resembles Z as much as possible. In this research, we model the topic as a distribution over either words or documents, depending on the applications. For word-level topic detection, the major task is to identify potential discriminative features for the latent topics. After that, by analyzing the correlations between discriminative features, a soft clustering would result in a set of topics, and each topic is described by a set of discriminative features. For document-level topic detection, the key idea is to find a suitable document representation scheme (or good support feature vectors) for measuring the similarities between documents, so that documents originating from the same topic are more similar to each other compared to documents from other topics. Other than topical level differences, the topic detection process can be further divided into online (real-time) and offline (batch) modes, which are also known as new event detection [APL98] and retrospective event detection [YPC98], respectively. Online topic detection incrementally examines each incoming news document to assess whether it belongs to an existing topic or if a new topic should be created based on it. Offline topic detection examines the entire corpus of news documents to simultaneously unravel topics and their associated news documents.

The granularity of a topic is hard to define. In general, a topic can be viewed as a kind of user preference, and can range from a broad topic like “earthquake” to a narrow topic like “some earthquake in North America with magnitude exceeding 8 ”. The topic resolution for the earthquake example is illustrated in Figure 1.2.

Figure 1.2: Different topic (user preference) resolutions for an “earthquake” topic.

Figure 1.2 illustrates the increasing difficulty in detecting relevant documents with respect to increasingly specific user preferences. Detecting whether an arbitrary topic will happen is beyond the capacity of traditional document categorization methods, which is geared towards clustering/classification of news topics. Instead of word-playing on the granularity of *topic*, we simply adopt the topic definition of Topic Detection and Tracking (TDT), where a topic is a cluster of documents initialized by the same seminal event, which has a well defined time and place [Tdt]. The recent Wenchuan Earthquake is one such example, whose initial event happened on May 12, 2008 in Sichuan province, China. The ensuing rescue related events lasted a couple of weeks, followed by the reconstruction events that span a few months. All documents related to these sub-events belong to the broader Wenchuan Earthquake topic.

Based on this topic definition, we shall not be concerned with any user preference that is more general than a topic. For example, “earthquake” can be a news category but not a topic, whereas the Wenchuan 512 earthquake, San Francisco 1989 earthquake, etc., are all

valid topics. On the other hand, we can delve deeper into a topic to identify a sequence of events, some of which may be more important than others. Rather than building the full structure sequence for all events in an arbitrary topic, which is an open research problem, we only detect the relevant documents that satisfy some user preferences. For example, for the Wenchuan earthquake topic, a user preference or event could be when the casualty figures exceed 60,000.

To sum up, in this research we are going to study three types of topic detection: word-level, document-level, and intra-topic.

1.3 Challenges and Our Contributions

From a data-mining perspective, offline and online topic detection may seem no different from offline and incremental document/word clustering, respectively. However, there are a number of subtle topic detection characteristics, which if not taken into due consideration, can adversely affect practical clustering performances: 1) time plays a pivotal role, with every news document bearing a time stamp, 2) news topics are naturally bursty, i.e., new topics are constantly generated while old topics die off, 3) news documents with semantically similar content but disparate time-frames most likely originated from different topics, e.g., hurricane Mitch of October 1998 and hurricane George of September 1998 are two distinct topics with considerable overlap in vocabulary. Topic detection can thus be viewed as a special case of stream clustering [GMM⁺03], with the clustering portfolio at any time point akin to a concept that drifts with time [Tsy04, HCL06b].

There are several challenges for topic detection, which will be addressed in this research, as described below.

- The hardest part of word-level topic detection is to find the proper correlations between word features. Traditional methods like latent Dirichlet allocation (LDA) [BNJ03] and word-level clustering [AHWY03] mainly focus on the document overlapping of words, without paying enough attentions to the role of the temporal dimension. In fact, the correlation between most features is non-stationary, and fluctuates significantly over time.
- Document-level topic detection methods suffer from the “curse of dimensionality” since hundreds of thousands of unique features must be considered. Documents in

very high-dimensional space are almost equally far away from one another, making classical similarity measures like Euclidean distances less discriminative. It is thus vital to identify temporally discriminative features for effective topic detection. Moreover, the vast majority of existing topic detection solutions [YPC98, ALJ00, SC01, All02b, YZCJ02, BCF03, KA04] do not explicitly incorporate time into their formulations; each news document is represented as a vector with time-agnostic static weights, with just one minor procedural modification: news vectors are processed in time-stamp order, i.e., online topic detection is used to handle the temporal factor. The entire setup smacks of ad hocism that builds upon the TFIDF (term frequency inverse document frequency) vector space model, which itself is lacking in mathematical formalism¹. Fortunately, this simple procedural modification over static text representation model seems to work quite well in practice. Other challenges include assigning documents into multiple topics, avoiding large amount of false alarms at low miss levels [AHF⁺05], etc.

- Detecting an arbitrary user specified events within a topic is even harder compared to the above two problems. Its ineffectiveness can be easily confirmed by subscribing to any of the many online news alerts services such as the industry-leading Google News Alerts [Gna], which generates more than 50% false alarms for any user preferences [HCL06b]. As further proof, portals like Yahoo take a more pragmatic approach by requiring all machine generated news alerts to go through a human operator for confirmation before sending them out to subscribers. This is because current state-of-the-art news alerts services use an unordered combination of keywords (that must or must not appear) to search for matching news. Clearly, this simplistic approach fails to grasp the finer semantics of a user-anticipated event.

In this research, we seek to tackle the above challenges mainly by investigating time-aware topic detection models. We summarize our main research contributions as follows.

- *Categorizing word feature trajectories via Discrete Fourier Transform*

To the best of our knowledge, we are the first to categorize word features for heterogeneous topics using frequency domain analysis of word feature trajectories. Specifically,

¹The reformulation of TFIDF via a probabilistic framework by Joachims [Joa97] does however help improve the standing of TFIDF.

every word feature is categorized into one of the following five feature types based on its power spectrum strength and periodicity: 1) HH (high power and high/long periodicity): important aperiodic topics, 2) HL (high power and low periodicity): important periodic topics, 3) LH (low power and high periodicity): unimportant aperiodic topics, 4) LL (low power and low periodicity): non-topics, and 5) SW (stopwords), a higher power and periodicity subset of LL comprising stopwords, which contains no information. We later found that word features from the HH, HL and LH quadrants can effectively serve as discriminative features for latent topics, and their power strength scores can be used to indicate the corresponding discriminative weights.

- *Offline and online topic detection via clustering discriminative word features*

We propose a simple and effective mixture density-based approach to model and detect offline word feature bursts. By monitoring the trend of each word feature’s dominant period, bursty word features can be identified. We also come up with an unsupervised word-level topic detection algorithm to detect both aperiodic and periodic topics for both online and offline news streams. Our algorithm has been validated on a real news stream and the results are very encouraging.

- *Discriminative probabilistic models for document-level offline and online topic detection*

The discriminative features identified by the trajectory analysis are further applied to document-level topic detection. We evaluate several time-aware probabilistic formulations for topic detection, and propose our own temporal and discriminative probabilistic framework called DIS. DIS is a Bayesian probabilistic framework that considers each news document as a point in discriminative word *and* time vector space. The posterior topic probability given a document *and* time is computed and subsequently used to assign a document to multiple fitting topics. One unique feature of DIS is that it only needs to operate on a subset of topic and temporal discriminative words instead of the full vocabulary space, making DIS very efficient for practical implementation. Moreover, formulating DIS using a probabilistic framework makes it more amenable to statistical analysis and practical deployment. For example, using a similar approach to [Joa97], we show that DIS is equivalent to a temporal version of TFIDF with discriminative feature selection, which can be trivially implemented. Lastly, a

thorough comparison against other generative probabilistic models is conducted, including offline and online topic experimental benchmarks using both soft and hard topic assignments. The soft topic assignment benchmarks are the first of its kind for topic detection, to the best of our knowledge.

- *Define and solve the problem of Anticipatory Event Detection (AED)*

We coin the term Anticipatory Event Detection (AED) to describe the task of detecting the occurrence of a user-specified anticipatory event within a topic. AED has a certain uncanny ability to analyze the intra-structure of topics. We propose approaching the AED problem by treating it as finding the transition between two adjacent events in an event transition graph whose events are represented by news articles covering the event transition graph before and after a particular transition has consummated. In practice, all topics of the same genre could share the same event transition graph. We subsequently define a sentence model and a document model for AED, treating the named entity types as the discriminative features for the event transition. The feasibility of AED is verified for some restricted domains including mergers and acquisitions and basketball news reports.

1.4 Thesis Organization

The rest of the thesis is organized as follows.

- In Chapter 2, we survey existing work from various fields related to this research. We first review the concepts and document representation in TDT. Then, the probabilistic models for topic detection, word-level topic detection and intra-topic analysis are discussed in details, respectively.
- In Chapter 3, we use techniques from signal processing to identify temporal discriminative features, which are bursty features, from a news stream. Two simple measures - dominant period and dominant power spectrum are used to classify each word into one of four quadrants, providing an adequate characterization in terms of the (a)periodic nature and forcefulness of the word. Unsupervised greedy algorithms based on bursty features are then developed for word-level offline and online detection of both aperiodic and periodic topics.

CHAPTER 1. INTRODUCTION

- In Chapter 4, we study non-probabilistic and probabilistic models for document-level offline and online topic detection. We proposed a temporal and discriminative probabilistic model called DIS, which is based on Bayesian probabilistic framework and built on the discriminative word features. Empirical comparisons with other more complex techniques such as vMF and LDA have been conducted to show the advantages of DIS in both offline and online settings.
- In Chapter 5, we define and propose solution for the Anticipatory Event Detection (AED) problem, which goes further into a topic by examining sequence of events. Supervised classification approaches are subsequently proposed to solve the AED problem by treating it as finding the transition between two adjacent events in an event transition graph (ETG).
- In Chapter 6, conclusions are discussed, followed by some directions for future work on discriminative word feature selection, probabilistic document-level models and AED.

Chapter 2

Related Work

We shall begin with an overview of existing research in areas related to topic detection. The motivation for this is two-fold. First, the areas reviewed will provide a better understanding of our discussion in the subsequent chapters. Second, to the best of our knowledge, the breadth of the literature covered also shows that the topic detection problem has never been addressed so comprehensively as done by us.

2.1 Background of TDT Community

Our research falls under the broader family of problems collectively known as Topic Detection and Tracking (TDT). More specifically, topic detection in TDT mainly refers to the document-level approach. In this section, we mainly introduce existing work originating from the TDT community, followed by how our work differentiate from others.

2.1.1 History of TDT

The first large-scale investigation on TDT started back in 1997 with a pilot study reported in [ACD⁺98]. Over the years, TDT gradually expanded to encompass the following areas:

- New event detection (NED) [APL98, YPC98, BCF03, KA04]: Detect the first document of a topic in an unsupervised manner without reference to any seed document.
- Topic detection (TD) [Car, YZCJ02]: Build clusters of news documents of the same topic.
- Topic tracking [FWMZ01, All02a]: Monitor a news stream to identify new documents belonging to an existing topic, which is defined by a few seed documents.

CHAPTER 2. RELATED WORK

- Story segmentation [BBL99, TSHTS01]: Detect changes between topically cohesive sections in each document.
- Story link detection [BPYC99, CFB03]: Determine whether two stories are topically linked.

The Corpora for system development and evaluation is provided by the Linguistic Data Consortium (LDC) [Ldc]. The LDC currently has five TDT corpora available for system development, the TDT Pilot study (TDT-Pilot), the TDT Phase 2 (TDT2), the TDT Phase 3 (TDT3), the TDT3 Arabic supplement, the TDT Phase 4 (TDT4) corpus, and the TDT Phase 5 (TDT5) corpus. In [Cie00], Cieri explained the methodology that the LDC has adopted for creating the reusable TDT corpora.

An overview of TDT research can be found in [All02b]. The overall goal of TDT is to understand news content across different languages, and to develop a system to process news streams from a variety of sources. The classical definition of *topic* from TDT 2004 is given as follows [Tdt],

Definition 2.1 (Topic) *A topic is a seminal event or activity, along with all directly related events and activities.*

Note that a TDT topic has a much narrower scope than traditional information retrieval (IR) topics or categories, and should be viewed more like a fine-grain news category. Likewise, a TDT *event* from TDT 2004 is defined as follows [Tdt],

Definition 2.2 (Event) *An event refers to a particular incident occurring at a specific time and place, along with all necessary preconditions and unavoidable consequences.*

The definition of a topic may be ambiguous at times. For example, “US attacked Iraq” can be a topic, whilst “US declared war with Iraq” could be a sub-topic or event inside since “US declared war with Iraq” can be further divided into events like “The F-117 made a surprise attack on Bagdad during the night”, etc. In fact, a hierarchical topic/event structure may gradually evolve [YS06, HCL06b]. Detection of multi-layers of relationships among topics or events is beyond the scope of this research. We will only consider a two-layer topic-event relation, and a flat structure of the topics/events in each layer, i.e., every topic/event is equally important. In addition, we shall adopt the TDT definition of topic/event, and equate new event detection with online topic detection. This definition naturally puts classical document clustering into the batch or offline topic detection genre.

2.1.2 Document Representation in TDT

As TDT deals mainly with news text documents, the first and foremost task is to transform document content into a representation suitable for modeling. The most commonly used document representation is the vector space model. In this model, each document is represented by a vector of words following the Vector Space Model (VSM) [SB88], where words are exchangeable (no orders). Given any collection of documents D , a word-by-document matrix $\mathbf{D}$ can be established to represent it based on VSM, where each entry represents the weight of a word in a document, i.e., $\mathbf{D} = (w_{ji})$, where w_{ji} is the weight of the j -th word in the i -th document. There are a number of ways to determine the weight w_{ji} , but most of them are based on two intuitive observations regarding text [AE99]:

- The more times a word occurs in a document, the more relevant it is to the document subject.
- The more times a word occurs throughout all documents in the collection, the more poorly it discriminates between documents.

The simplest weighting approach is binary encoding, which sets the weight to 1 if the word occurs in the document and 0 otherwise. Another approach uses the frequency of the word in the document, i.e., $w_{ji} = f_{x_j, d_i}$ which measures the term frequency of word feature x_j in document d_i . A more common weighting approach is the TFIDF (term frequency - inverse document frequency) weighting:

$$w_{ji} = f_{x_j, d_i} \times \log\left(\frac{N}{N_{x_j}}\right), \quad (\text{Eq. 2.1})$$

where N is the number of documents in the corpus D and N_{x_j} is the number of documents for which the word feature x_j exists in D . Various TFIDF weighting variations have been proposed for different applications. To eliminate bias towards lengthy documents, a popular smoothed variation of TFIDF which takes into account the length of document was given by [Kwo98] as follows:

$$w_{ji} = \frac{f_{x_j, d_i}}{\sqrt{\sum_{k=1}^n f_{x_k, d_i}^2}} \times \log\left(\frac{N}{N_{x_j}}\right). \quad (\text{Eq. 2.2})$$

For matrix $\mathbf{D}$, the number of columns corresponds to the number of words n in the document collection. As there could be hundreds of thousands of different words, stopword (frequent word that carries no information) removal, word stemming (suffix removal), additional

dimensionality reduction techniques such as Principal Component Analysis (PCA) [I.86] and Fisher’s Discriminant Analysis (FDA) [Fis36], feature selection, and re-parameterizations [AE99] are usually employed.

2.1.3 Similarity Measurement in TDT

Similarity comparison is a core technique used in TDT. The cosine similarity is the most popular similarity measurement in TDT, which is defined between two sample vectors as follows,

$$\text{cosine}(\mathbf{d}_i, \mathbf{d}_j) = \frac{\mathbf{d}_i \cdot \mathbf{d}_j^T}{\|\mathbf{d}_i\| \|\mathbf{d}_j\|}, \quad (\text{Eq. 2.3})$$

where $\|\mathbf{d}\| = \sqrt{w_1^2 + \dots + w_n^2}$ is the $L2$ norm of $\mathbf{d}$.

The other measurements include a symmetric Okapi formula defined by Robertson *et al.* [RWHB⁺96] which is given by

$$\text{okapi}(\mathbf{d}_i, \mathbf{d}_j) = \sum_{x_k \in X, w_{ki} \neq 0, w_{kj} \neq 0} f_{x_k, d_i} f_{x_k, d_j} N_{x_k}. \quad (\text{Eq. 2.4})$$

A normalization constant is often included so that $\sum_{x_k} f_{x_k, d} = 1$ and independent of the document length. Other normalization methods uses L_2 norm [SL99] and Gaussian modeling of scores [JSSW99, WJSS99]. In [FWMZ01], Franz and McCarley updated Okapi formula with respect to cluster membership c as

$$\text{okapi}(\mathbf{d}_i, \mathbf{d}_j, c) = \sum_{x_k \in X, w_{ki} \neq 0, w_{kj} \neq 0} f_{x_k, d_i} f_{x_k, d_j} N_{x_k}(c), \quad (\text{Eq. 2.5})$$

where $N_{x_k}(c)$ is the number of documents for which the word feature x_k exists within cluster c in the corpus D .

Another important improvement in the performance of TDT is “dynamic word weight” discussed in [FWMZ01, DFM⁺99b, DFM⁺99a]. The “dynamic word weight” modifies the inverse document frequency based on the following assumption: given a document d and a feature x from it, if x is contained by a lot of other documents, its weight is intuitively diminished; however, if most of such documents come from the same cluster as document d , the weight of x is increased. One formulation of “dynamic word weight” is defined in [FWMZ01] as follows

$$N'_x(c) = N_x + \beta \frac{2N_x(c)}{N_x + N(c)}, \quad (\text{Eq. 2.6})$$

where $N(c)$ is the number of documents clustered into the cluster c in the corpus D . The document corpus and the cluster c dynamically change coverage over time, thus this “dynamic word weight” is time dependent. Other cluster based extensions in “dynamic word weight” can be found in [HGM00, YAPL00].

Other document-document similarity approaches in the TDT community include the conditional probability model of generating a new document [YCG⁺99, S.A99], similarity functions and terms weights from well-known IR system like INQUERY [APL98], SMART [YPC98], PRISE [Oar99], and linguistically motivated feature selection [HGM00].

2.1.4 Topic Detection in TDT and Clustering

Most of the existing TDT solutions modeled topic detection as a non-probabilistic clustering problem, and have achieved significant success in the past. For example, Yang *et al.* [YPC98] successfully used hierarchical clustering for offline topic detection. Allan *et al.* [AHF⁺05] showed that Single-Link and Single-Pass clustering, which assigns the cluster label of the nearest neighbor (1-NN), achieved the best online topic detection performance. In fact, if we only consider a flat topic structure and ignoring time, there exists extensive work on batch or offline topic detection, including the rich family of k -means clustering [DJ88, KR90, DM01].

We categorize these TDT solutions as “non-probabilistic clustering models” or “similarity-based methods”. More specifically, the details of each approach is described as follows.

In [APL98], Allan *et al.* conducted a TDT pilot study in which they defined various problems including new event detection (NED) and event tracking (similar to topic tracking). They also proposed a feature selection method by identifying “surprising” (infrequent) features, and developed a solution of event tracking based on a few training stories. In [AJR⁺99], Allan *et al.* summarized the state-of-the-art research presented at the 1999 TDT workshop, and also clearly defined the scope and standardized evaluation methodologies for various TDT problem domains. In [JSSW99], Jin *et al.* used an incremental k -means algorithm to cluster the news articles into various topics before performing TDT. In [ALJ00], Allan *et al.* empirically showed that NED is a very hard problem using only cosine similarity based approaches. In [ALMS00], in the summarization of the system used by University of Massachusetts for TDT tasks, Allan *et al.* further explained that first story detection based on the tracking methods is unlikely to be efficient. In [Car, CS00], Carthy *et al.* implemented two topic tracking systems based on traditional IR techniques

CHAPTER 2. RELATED WORK

and using the lexical chaining respectively. They applied the lexical chain to discover the co-reference of words in one sentence and claimed that the second system with the help of the lexical chain outperformed the keyword-based system significantly. In [FWMZ01], Franz and McCarley firstly represented a full conversion of NED into topic tracking by replacing the document-document similarity with the similarity between a document and a cluster centroid. In [SC01], Stokes *et al.* proposed a composite document representation using both lexical chains and proper nouns for NED. We believed that their method is actually a variation of applying named entities in NED. In [YZCJ02], Yang *et al.* claimed to achieve a substantial performance gain by first classifying news articles into different topics, followed by applying 1NN to detect new events (NED). Their result agrees with first principles since a similarity comparison within news events of the same topic should work well to identify the outliers (new event). However, they did not attempt classification directly to detect new events. In [BCF03], Brants *et al.* used a combination of techniques, including Hellinger distance, tiling for a better documents matching, etc. on NED and claimed to improve the overall performance significantly. From [BCF03], we see that numerous attempts have been made to improve NED, all without spectacular results. It appears that a performance breakthrough would require a higher order understanding of the problem domain. Thus, we believe that unless domain knowledge is utilized, NED will remain a hard problem for the foreseeable future.

In [KA04], Kumaran *et al.* used text classification techniques and named entities, but for detecting *all* new events of a particular category (using a model trained threshold). They found that the financial category is the only category in which the function of named entities cannot be determined, and named-entities-only representation worked better for some categories (e.g. Legal and Science) and non-name-entity-terms-only representation worked better for the other categories (e.g. Elections and Sports). In our simulated cosine similarity methods for AED, named entities were found to work consistently with their findings, as our testbed comprises primarily financial news articles. However, for our AED model, we found that the representation of named entity types along with non-named entity terms work better than both named-entities-only representation and non-named-entity-terms-only representation.

Retrospective event detection (RED), or offline topic detection, was usually modeled as a offline clustering problem. Li *et al.* [LWLM05] attempted to identify topics/events

within a corpus of historical date-stamped news articles with the help of both time and content information. It assumes that the news topic histogram of a particular topic genre are Gaussian-distributed with peaks/bursts denoting a new topic.

The studies on TDT also involve multi-languages except for English, such as in Serbo-Croatian [GFS⁺98]. Two main differences from English are: 1) the lack of available acoustic and language data, and 2) the complicated syntax and excessive vocabulary growth.

2.1.5 Novelty of Our Research

Our work on document-level topic detection continues to use the vector space models and the cosine similarity widely adopted in TDT. However, our work is different from the vast majority of TDT approach in that 1) we explicitly include time in our model; 2) we focus on the family of probabilistic topic detection models. The word-level topic detection and AED have not been studied by the TDT community. In the following sections, we shall review related work on probabilistic models, word-level topic detection, and AED.

2.2 Probabilistic Models for Topic Detection

There exists a plethora of literature on probabilistic models for unsupervised text clustering, most of which can be directly applied to offline topic detection, ignoring the time factor. We first introduce the Naive Bayes model, which is fundamental to the other advanced probabilistic models like vMF mixture, LDA and our DIS model as follows.

2.2.1 Naive Bayes Model

A Naive Bayes model [Joa97] was first proposed as a simple probabilistic classifier, which can be naturally extended for unsupervised topic detection as we did in our work. “Naive Bayes models are based on probability models that incorporate strong word independence assumptions which often have no bearing in reality, hence are naive. [Wik]” A more fitting term for the underlying probability model would be the independent feature model. The posterior probability of a class z given a document d can be derived using Bayes’ theorem as

$$P(z|\mathbf{d}) = \frac{P(z)P(\mathbf{d}|z)}{P(\mathbf{d})}, \quad (\text{Eq. 2.7})$$

where $P(\mathbf{d})$ does not vary across categories and can be left out. Based on the assumption of word independence, the above equation is simplified as

$$P(z|\mathbf{d}) = P(z) \prod_{k=1}^n P(x_k|z) \quad (\text{Eq. 2.8})$$

An estimate of the prior $P(z)$, $\hat{P}(z)$, is given by the fraction of training documents belonging to class z as

$$\hat{P}(z) = \frac{N(z)}{N}, \quad (\text{Eq. 2.9})$$

and the estimate of $P(x|z)$, $\hat{P}(x|z)$ is given by

$$\hat{P}(x|z) = \frac{1 + N_x(z)}{n + \sum_{k=1}^n f_{x_k, z}}, \quad (\text{Eq. 2.10})$$

where $f_{x_k, z}$ is the overall frequency of the word feature x_k in the documents for which the true class is z . The assigned topic is achieved through the maximum likelihood. For example, given a document d , we compute its class-posterior probabilities $P(z_k|\mathbf{d}) \forall k = 1, \dots, K$, and it is assigned to the class z' that has the highest class-posterior probabilities, i.e.,

$$z' = \arg \max_z P(z|\mathbf{d}). \quad (\text{Eq. 2.11})$$

Although the assumption of feature independence is generally not true, Naive Bayes models are surprisingly accurate on many datasets and fairly popular due to the fact that the ultimate topic assignment does not depend on the estimated posterior probabilities but only on their relative rankings. In fact adding explicit correlations between features usually makes things worse, due to the difficulty in reliably estimating joint probabilities of two or more dimensions.

2.2.2 Generative Probabilistic Models

Generative probabilistic models consider topics as distributions of either documents or words. Models that use the vector space word representation of documents typically model a topic as a distribution of documents, i.e., the Gaussian mixture model [Das99] and von-Mises Fisher (vMF) mixture model [BB07], each of which models a topic as a Gaussian or vMF distribution of documents in word vector space, respectively. Other probabilistic models such as naive-Bayes [NMTM00], Dirichlet Compound Multinomial (DCM) mixtures [Elk06], probabilistic Latent Semantic Indexing (pLSI) [Hof99], and Latent Dirichlet

Allocation (LDA) [BNJ03] treat each topic as a multinomial distribution of words. We can also simplify the topic distribution by modeling each topic as a discrete probability over documents, which is exactly what our DIS model assumes.

Most of the probabilistic models follow the Naive Bayes assumption of feature independence while estimating the posterior probability $P(z|\mathbf{d})$. The main differences lie in whether the topic variable z is a distribution over document set D or feature set X . For example, the mixture probabilistic models assume k topics, each of which is some distribution Θ (i.e., Gaussian or vMF) over all documents. Therefore, given one distribution Θ , we can generate the probability of $P(\mathbf{d}|\Theta)$ depending on how d is close to the mean of Θ . Accordingly, among all k topic distributions, one distribution having the highest generative probability on d is then assigned as the topic label to d . Both document and topic variables are represented by the vector space models, where word features are independent and exchangeable to each other. For word-level topic models, the document generative probability $P(\mathbf{d}|\Theta)$ is further expanded to

$$P(\mathbf{d}|\Theta) = \prod_{k=1}^X P(x_k|\Theta), \quad (\text{Eq. 2.12})$$

where each word feature's generative probability is calculated independently. Certain topic distribution Θ (i.e., multinomial) must be assumed. Each kind of probabilistic model has its own specific advantages, but their functions on topic detection was not obvious until now. We shall study this problem in greater details in Chapter 4.

There are but a handful of probabilistic models specifically geared for online topic detection, which as we emphasized before, depends strongly on time. Moreover, simple online extensions of vMF mixture and LDA have been proposed in [BB07] and [BL06] respectively, where the focus has been on incrementally updating existing cluster topic parameters (assuming a fixed number of clusters), rather than detecting new topics. Zhang et. al. [ZGY05] were the first to apply LDA to online topic detection. However, training on existing known topics is needed to model a new topic prior, which makes this approach supervised.

2.2.3 Novelty of Our Research

Our work on document-level topic detection has fundamental differences from the above generative probabilistic models: 1) we proposed a temporal discriminative probabilistic model

DIS by removing the distribution assumptions made by other more sophisticated probabilistic models and incorporating the time element; 2) we comprehensively benchmarked various probabilistic models including DIS on both offline and online topic detection using both hard and soft topic assignments, making our work one of the most comprehensive topic detection benchmarks to date.

2.3 Word-level Topic Detection

Observe that the temporal dimension has received little attention in the past. One probable reason for this is that most of the existing work consider document-level retrieval; a document is but a one-shot occurrence with a time-stamp, with no continuity in time. On the other hand, if we look at words across documents and time, each word can actually be modeled as a time series, where each element can be any statistical value, e.g., document frequency at one time point. Swan and Allan [SA00] first attempted to reconstruct an event by propagating the temporal relationships among co-occurring terms. However, their work were limited in named entities and noun phrase pairs, and did not differentiate between the various feature genres, i.e., bursty and non-bursty. The word-level generative probabilistic models introduced in Section 2.2.2 have certain ability of generating topics over words, however, they are in general not practicable due to a few reasons: 1) the number of topics has to be pre-defined; 2) the running time is very expansive; and 3) their online versions are simply interval offline clustering, losing the temporal continuity of word trajectories. Instead, we focus on the bursty behaviors of word trajectories, and then develop some general (compared to [SA00]) methods with less assumptions and faster processing speed (compared to generative probabilistic model). In the following existing work related to burstiness analysis on topic detection is briefed.

2.3.1 TDT Approaches Utilizing Bursts

Recently, the study on “burst” for event detection has garnered a lot of interest. The pioneering work of Kleinberg [Kle02] modeled an event in text streams as a “burst of activities” using word features. Each word feature is assumed to be in one of two states, *bursty* and *non-bursty*, and a 2-state finite automaton model was designed to determine its bursty periods and corresponding bursty weights. Ideally, a group of features sharing similar bursts indicates the same event. However, this pilot study suffers from two practical

limitations. First, there is no systematic way to determine the boundary between *bursty* and *non-bursty* states. A high value leads to only a few bursts, but a low value results in many useless bursts from frequent yet meaningless features, i.e., stopwords. Second, this work applied dynamic programming to differentiate bursty and non-bursty features, restricting its application to historical data.

Some follow-up work since then have attempted to tackle the above limitations. Fung *et al.* [FYYL05] modeled each burst as a binomial distribution, which associates a continuous bursty probability to each feature over time. Zhu and Shasha [ZS03] detected online bursts by modeling segments of streaming data within elastic sliding windows as Shifted Wavelet Trees. Trees (segments) with significantly different aggregates are extracted as bursty periods. Kifer *et al.* [KBDG04] further extended the burst detection from discrete data streams (text streams) to continuous streams by measuring the distances between two arbitrary distributions, each corresponding to two consecutive sliding window/period.

We categorize all of the above work as “first-order methods” because they all measure bursts using some sort of first-order statistics over a (moving) average window. Although all of them have reported notable results, there are still some issues to be addressed. For example, [FYYL05] bypassed the training of the boundary between *bursty* and non-bursty states, but instead added additional assumptions, i.e., all bursty periods are determined by the binomial distribution, within which each bursty weight follows a sigmoidal distribution. To remove such constraints, in this research we measure the degree of burstiness with its power spectral density so that all bursts can be naturally weighted and ranked. The approaches of [ZS03] and [KBDG04] only examine the statistical difference between two (consecutive) sliding windows, which is highly susceptible to noise, including false bursts. In this research, we identify bursts with a stable frequency, thereby taking care of any temporary fluctuations.

The idea of ‘burst’ also inspired a series of other applications such as Kumar’s identification of bursty communities from Weblog graphs [KNRT05], Mei’s summarization of evolutionary themes in text streams [MZ05], He’s work in clustering text streams using a combination of binary and bursty features [HCLZ07], etc.

2.3.2 Trajectory Analysis of Bursty Phenomena

Electrical engineers have been analyzing temporal signals for many decades, and have devised various useful formulations, among which include Fourier and wavelet transforms [Mal99].

Although these techniques have become de facto standards in engineering, they have rarely been applied to text streams with the exception of the pioneering work of Vlachos *et al.* [VMVG04]. They used spectral analysis techniques to identify periodicities and bursts from query logs. Their focus was on detecting multiple periodicities from the power spectrum graph, which were then used to index words for “query-by-burst” search. In this research, we also make use of spectral analysis, but with two major difference from their work, 1) we apply it on news streams to simultaneously classify word features along two dimensions, namely periodicity and power spectrum, 2) our ultimate goal is to reconstruct interesting periodic and aperiodic events from a subset of bursty words.

Compared to [VMVG04], we are only interested in the dominant (largest DF, or highest energy) frequency of each word trajectory. Therefore, Fourier transforms, which are agnostic to the temporal localization of spectrum components, suffice for our purpose. Further, we actually assume that the dominant word frequency is valid over the entire duration of the news stream. This assumption is natural since an aperiodic event is equivalent to a “periodic” event with periodicity equals to the entire duration, for all practical purposes. Of course, our approach would not identify the local periodicities of events that are locally periodic. To detect multiple periodicities, a more sophisticated transform such as wavelet analysis could be used, which is beyond the scope of this research.

2.3.3 Novelty of Our Research

We extensively study the heterogenous features and their related heterogenous topics by incorporating the time and frequency information for the first time. To the best of our knowledge, our approach is also the first to conduct word-level topic detection via the burstiness analysis for word feature trajectories.

2.4 Intra-topic Analysis

Except for topic detection and tracking tasks, in this research we also conducted intra-topic analysis: within a topic, how events evolve over time. Learning concept drift of a time varying source is not new in classical fields like signal processing and machine learning, where researchers mainly focused on the incremental learning process [BR82, SRHG86, KPR90, Kli04] that automatically adapts a model to handle concept drifts. However, in TDT, users are more interested in the occurrence of their interested/subscribed events, not just any

changes. For example, Google News Alerts allows users to specify a few keywords as the event transition condition, for which they will be notified if the keywords are found in new documents. Therefore, instead of implicitly learning the concept drift, in this research we explicitly used the transition condition (can be viewed as a concept drift) to learn a binary classification model for each event transition. More specifically, we conducted intra-topic analysis by defining the problem of Anticipatory Event Detection (AED). We defined an Event Transition Graph (ETG), otherwise known as *Event Evolution Graph*, as a directed graph to model a set of events within a specific topic. Specific states (events) of the topic are represented by nodes, with the edges denoting possible transitions and associated firing conditions. In general, events from the same topic genre (i.e, earthquake) can share a common ETG. Some work related to AED are described below.

2.4.1 Event Transition Graph (ETG)

Makkonen [Mak] coined the term *event evolution* to denote the various time stages of a typical topic. In this work, an event is comprised of time-ordered related documents, and multiple events together constitute an event evolution graph. Nallapati *et al.* [NFPA04] defined *event threading* as the dependencies between events. They viewed the news event as a relational structure. They used interdependencies between news events to build a unsupervised relational structure, which is somewhat similar to building the *event transition graph* in AED, since they also studied the semantics of news events. However, their focus lies in building the news event relational structure. At least two main differences to our work are as follows: 1) without the help of domain knowledge, finding the structure of news event is nothing more than a clustering problem. There is no guarantee of generating a sensible structure capable of representing real event relations. 2) their ultimate goal is to solve NED; after the news event is split into several news threads, they then try to detect the first story of each thread using cosine similarity. Therefore, the main idea of their work is to partition an existing topic into smaller conceptual events. Yang *et al.* [YS06] formally defined event evolution as the relationship between all events within a topic. Each relationship is in fact a transition that traverses in time from seminal events to terminal events. Their work evaluates the similarity between events, which are assumed to be comprised of well-clustered documents, to derive candidate transitions.

In our work, AED also assumes events to be a time-ordered sequence of documents. However, we make a simplifying assumption that a pre-existing 2-state ETG is readily available. We have yet to address the challenges of building an ETG from scratch. In principle, AED could use a combination of the above techniques to come up with a reliable ETG based on historically similar (to the current anticipatory event) events.

2.4.2 Domain Knowledge Based Topic Detection

ETG is nothing more than a time-ordered domain knowledge for a specific topic genre. There are some other work that has tried incorporating domain knowledge to tackle the TDT problem [FS00, MLRM04]. However, none of the existing work attempts to construct an ETG as in our case, which is important for representing relationships between events. In [FS00], Fukumoto and Suzuki made use of the domain dependency of word frequencies to weigh each word according to its discriminatory power to discern a topic from the others. They also applied the scheme to track broadcast news articles within the same topic. Their work was not aimed at identifying the physical relationships between events within a topic because they only statistically single out the strong domain or event-genre related words. In [MLRM04], by dynamically extracting topic descriptors and discriminators to search for novel information, Maguitman *et al.* clustered news articles into topics within an initial conceptual map. This is an intuitively reasonable application of applying the domain knowledge (the concept map) on topic detection. Moreover, this introduced yet another open problem, that of the accurate extraction of descriptors/discriminators. Consequently, we feel that the direct application of domain knowledge to topic detection would fare better as errors due to incorrect knowledge representation would not be amplified.

2.4.3 Personalized TDT

AED can also be viewed as a personalized TDT problem since only specific events interesting to the user would be detected. Previous research on personalized TDT include [BGMV04, GDH04]. In [BGMV04], Bengel *et al.* attempted to use traditional classification techniques to create the conceptual profiles from chat records. However, their work largely resembles a search engine specialized for chat room content. The combination of pre-trained topic models and individual chat profiles are used to improve/filter search results. In [GDH04], Gabrilovich *et al.* personalized news alerts by presenting only the most novel news articles

according to a user profile. Distinguishing novelty from noise is a tricky endeavor, since the majority of novel (different) news articles could be irrelevant, with only a few that are actually related to the events.

In [AGK01a, AGK01b], Allan *et al.* extracted a single sentence from all news articles over time for each event within a topic. Their goal is to monitor the changes in events covering a topic. They calculated two probabilities for each sentence, probability of *usefulness* (measurement of on-event hit of a sentence) and probability of *novelty* (measurement of novelty with respect to prior sentences for a sentence), and a joint probability is computed for each sentence. The sentence with the highest joint probability will be the summary of an event. They used all prior sentences to estimate the joint likelihood that all different words will appear in one sentence independently, as its probability of *usefulness*. Another model to estimate the probability of *usefulness* for one sentence is that if it comes from a predominant part of the news article, it will be assigned more weight in its *usefulness*. To compute the *novelty* for one sentence, they compared “its” event with all the prior sentences. Another option was to cluster the sentences in the same event and then compare “its” event with all the prior cluster means. They claimed that finding the *usefulness* is hard but good approaches for finding the *novelty* to avoid redundancy can be derived. They also mentioned it is not trivial to effectively combine *usefulness* and *novelty*.

2.4.4 Novelty of Our Research

We semantically distinguish one event from an adjacent event within an ETG, with the final goal of firing user anticipated events on-the-fly. We thus define and study the novel AED problem which is highly useful in practice despite its simpleness. Understanding the intra-structure of a topic genre is highly dependent on domain knowledge. Although there exist some existing work with overly ambitious goals, they are practically infeasible.

2.5 Other Topic Detection Research

There are some other work related to the general topic detection yet not directly related to our research. We briefly describe them as follows.

Some researchers improved the TDT performance via user feedback. Raghavan and Allan [RA05] used explicit user feedback to make topic detection tasks more practical.

They allowed user to point out relevant passages in addition to specifying relevant documents. Zhao *et al.* [ZLBM06] applied web page click-through information as the implicit user feedback to detect events that share similar evolutionary pattern over time, and also achieved decent performances with the limitation that most text streams do not have such click-through information.

Some researchers tried to improve the document representation by embedding more semantics using Natural Language Processing (NLP), which is the study of automated generation and understanding of natural human languages. In [HSC00], Hatch *et al.* aimed to solve TDT research problems by introducing lexical chains and conceptual indexing under the WordNet framework. A lexical chain is a group of semantically related words responsible for the context behind a text fragment. The lexical chain was first formulated by [Has84, HH76]. The WordNet thesaurus software [MI90] was applied as a lexical resource in [HSC00] to cluster words by meanings. Accordingly, they built a conceptual indexing model by representing a document with two vectors: one of which contains the words found in WordNet noun file and their synonymous words from WordNet noun file, and the other contains other words not found in the WordNet noun file. After conceptual indexing, they were able to equally interpret one article about “cars” and another article about “automobiles”. However, an obvious disadvantage is that the word sense disambiguation brought much more irrelevant words into the document representation.

2.6 Summary

In this chapter, we reviewed a large amount of existing work in TDT. Document-level topic detection has been well studied as a non-probabilistic problem in the past, yet word-level topic detection and intra-topic analysis have not been tackled thoroughly. Moreover, the time dimension has been largely ignored. The ensuing chapters provide an in-depth study of these issues.

Chapter 3

Discriminative Features of News Streams and Word-Level Topic Detection

In this chapter, we describe how discriminative features for latent topics can be extracted from a news stream. The discriminative features are subsequently used for topic detection.

3.1 Motivations of Discriminative Features

The motivation of the work in this chapter stems from the observation that the vast majority of topic-aware applications require only a few crucial keywords to distinctively represent each topic, rather than a bunch of documents. For example, an event search engine may simply use a bunch of event-specific keywords to index every event. Therefore, an unsupervised solution to automatically recognize discriminative features for topics should be extremely useful. Another example is the news alerts system, which notifies subscribers upon the occurrence of one or more user-specified anticipatory topics/events. Unfortunately, current subscribers to such systems receive a large amount of false-alarm documents [HCL06b]. In fact, to alleviate this problem, a number of news alert providers like Yahoo had to resort to hiring human operators to filter and summarize all machine generated news alerts before sending them out to the subscribers. Ideally we would like to be able to extract discriminative features from every significant topic genre (e.g., earthquake), which can be used to predict subsequent topics of the same genre effectively.

Although TDT approaches have been studied extensively for the past 10 years, we have not been able to achieve the aforementioned ideal due to several reasons. First,

not much work has been done on recognizing discriminative features for topics in a completely unsupervised manner. The vast majority of TDT solutions proposed for topic detection [YPC98, ALJ00, SC01, All02b, YZCJ02, BCF03, KA04, HCL06b] is based on clustering [ALJ00] of document vectors. Second, we are unaware of any authoritative work on word-level online topic prediction. The need to tune a large number of parameters in most existing methods [AHF⁺05, FYYL05] in fact prevents us from detecting online topics. Third, the temporal aspect, which plays a crucial role in topic detection has been largely ignored over the years. Despite the fact that the evolutionary trends of discriminative features over time can provide valuable domain knowledge for detecting new topics, a lot of existing work [KA04] in the field only uses time to order the documents.

Instead of attacking the problem with variations of the same hammer (cosine similarity and TFIDF), we seek a fundamental understanding of word feature trajectories in both time and frequency domains. Our previous work [HCL07] has attempted to analyze the historical trajectories with the goal of summarizing retrospective topics. In this research, we further extend that approach to online topics. To better tap into our idea, several important observations are illustrated as follows.

First, in all topic detection tasks, there is a large set of candidate word features only for a fixed set of observations, i.e., obvious bursts. Among these features, only a small number are expected to be useful, and only over a certain period of time. For example, consider the evolutionary trends of the words *Easter* and *April* from the Reuters Corpus¹, whose normalized DF (Document Frequency, which is the number of documents containing that word) trends are plotted in Figure 3.1.a. From the plot, both words clearly exhibit a bursty

3.1.a: aperiodic topic

3.1.b: periodic topic

Figure 3.1: Feature correlation (Normalized DF versus time) between a) *Easter* and *April* b) *Unaudited* and *Ended*.

¹Reuters Corpus is the dataset used in this chapter.

trend *around* the time of the *Easter feast* (04/1997). This means that among hundreds of thousands of word features, *Easter* and *April* are probably the two most discriminative words for the topic *Easter feast*. From the same figure, we can also see the strong correlation between the two words.

The second observation comes from the frequency domain. Consider the example shown in Figure 3.1.b, where the words *Unaudited* and *Ended* exhibit similar behavior over periods of 3 months. That is to say, not only do these two words possibly relate to the same topic, but also both the topic and all of its discriminative features have a period of 3 months. Interestingly, the two words actually originated from the same periodic topic, *net income-loss reports*, which are released quarterly by publicly listed companies. Similarly, in the previous example, the underlying topic *Easter feast* is a typical important aperiodic topic with respect to our 1-year data, so that its discriminative features (*Easter* and *April*) also have a minimum period of 1 year².

Other observations drawn from Figure 3.1 are: 1) the bursty period of *April* is much longer than *Easter*, which suggests that *April* may exist in other topics during the same period; 2) *Unaudited* has a higher normalized DF value than *Ended*, which indicates *Unaudited* to be more discriminative for the underlying topic.

These two examples are but the tip of the iceberg among all word trends and correlations embedded in a news stream like Reuters. If a large number of them can be uncovered, it could significantly aid TDT tasks. In particular, it illustrates the importance of mining correlated features in both time and frequency domains for topic discovery.

We thus propose a method to analyze word trajectories, and subsequently use that information to accurately identify both offline and online topics in the following sections. Our methods postulate that: 1) A topic is described by its discriminative features. A periodic topic has a list of periodic features and an aperiodic topic has a list of aperiodic features; 2) Discriminative features from the same topic share similar distributions over time and are highly correlated; 3) An important topic has a set of active (largely reported) discriminative features, whereas an unimportant topic has a set of inactive (less-reported) discriminative features; 4) A feature may be included by multiple topics with overlaps in time frames.

²Since our data only spans 1 year, without additional domain knowledge, the most the system can conclude is that the topic is either aperiodic, or it has a periodicity of 1 year or longer.

3.2 Problem Definition for Word-Level Topic Detection

Word-level Topic Detection aims to find the set of ordered discriminative word features for each latent topic from a temporal text stream. The time duration T of a news stream is fixed for a historical stream but countably finite $(+\infty)$ for an online source. We also define T_c to be the current time point for online topic detection.

Naturally, not all word features are discriminative with respect to a topic. For example, the words *hurricane*, *sweep*, and *strike* could all be discriminative features of a Hurricane genre topic. Likewise, a set of strongly correlated features could be used to reconstruct a topic description. Accordingly, we define *discriminative feature* as follows.

Definition 3.1 (Discriminative Feature) *If a word feature uniquely discriminates a topic from all others, it is called a discriminative feature for that topic.*

Note that the choice of *discriminative features* can be rather subjective and varies for different datasets.

Within T , there may exist certain topics that occur only once, e.g., *Tony Blair elected as Prime Minister of U.K.*, and other recurring topics with varying periodicities, e.g., *weekly soccer matches*. We thus categorize all topics into two types: aperiodic or periodic, defined as follows.

Definition 3.2 (Aperiodic Topic) *A topic is aperiodic within T if it only occurs once.*

Definition 3.3 (Periodic Topic) *If topics of a certain topic genre occur regularly with a fixed periodicity $P \leq \lceil T/2 \rceil$, we say that this particular topic genre is periodic, with each member topic qualified as a periodic topic.*

Note that the definition of “aperiodic” is relative, i.e., it is true for a given T , and may be invalid for any other $T' > T$. For example, the topic *Christmas feast* is aperiodic for $T \leq 365$ but likely to be periodic for $T \geq 730$. Such situations are more prevalent for online data, where $T \in (0, +\infty)$. We can therefore only effectively qualify any detected aperiodic topics up to T_c .

Except for the periodicity classification, topics can also be categorized into important (largely-reported or popular) topics and unimportant (less-reported or rare) topics.

Given a news stream, our first target is to filter all discriminative features from the non-discriminative ones. Then, we categorize discriminative features with respect to the categories of topics so that each set of discriminative features corresponds to one type of topics. After analyzing the correlations among discriminative features within the same category, we then classify them into various discriminative feature groups, each of which might portray one topic.

3.3 Modeling Feature Trajectory

In this section we first introduce the basic feature trajectory vector model. Next we define some bursty feature concepts. Finally, spectral analysis is applied onto each feature trajectory to quantify its burstiness.

3.3.1 Feature Trajectory Vector

Suppose we track a word feature such as DFIDF over time, and order its daily values into a trajectory vector, which essentially becomes a time series signal in engineering parlance.

For offline and online data, we have slightly different trajectory representations. We are able to measure some statistical information such as the mean over the entire time duration for a historical dataset. However, we can only compute a moving average for online data. We first define the offline representation vector for a word feature as follows.

Definition 3.4 (Offline Feature Trajectory) *The offline trajectory of a word feature x can be written as the sequence*

$$y_x = [y_x(1), y_x(2), \dots, y_x(T)],$$

where each element $y_x(t)$ is a measure of feature x at time t , which could be defined using the DFIDF score

$$y_x(t) = \frac{N_x(t)}{N(t)} \times \log\left(\frac{N}{N_x}\right), \quad (\text{Eq. 3.1})$$

where $N_x(t)$ is the number of documents (local DF) containing feature x at day t , N_x is the total number of documents (global DF) containing feature x over T , $N(t)$ is the number of documents for day t , and N is the total number of documents over T .

Likewise, the online representation vector for a word feature is defined as

Definition 3.5 (Online Feature Trajectory) *At the current time $T_c \geq 1$, the online trajectory of a word feature x can be written as the sequence*

$$y_x = [y_x(1), y_x(2), \dots, y_x(T_c)],$$

where each element $y_x(t)$ is a measure of feature x at time t , which could be defined using the moving DFIDF score

$$y_x(t) = \frac{N_x(t)}{N(t)} \times \log\left(\frac{\sum_{t'=1}^{T_c} N(t')}{\sum_{t'=1}^{T_c} N_x(t')}\right),$$

where $N_x(t')$ is the number of documents (local DF) containing feature x at day t' , $N(t')$ is the number of documents for day t' .

Note that each element of the online feature trajectory vector dynamically changes over different T_c , because the IDF part varies on a daily basis with t .

3.3.2 Bursty Features

We model a topic in text streams as a “burst of activities”, where the activities are represented by topic features. Therefore, for a topic detection task we are only interested in features that clearly exhibit bursty behavior, defined as *bursty feature*. We shall ignore the remaining less obvious features that our algorithm fails to detect, and treat them as noise. Although *bursty features* are only approximations of *discriminative features* for latent bursty topics, later on we will show that they are quite effective for topic representation and detection.

The burstiness behavior has been extensively studied for similar time series in network traffic and control theory [Spi82]. However, instead of defining the burstiness as Peak-to-Average ratio, we follow a *burstiness* definition resembling the *bursty factor* of [Lam78] as follows.

Definition 3.6 (Burstiness) *The burstiness of the word feature x within the time range $[T_l, T_u]$ is measured by*

$$b_x = \frac{N'_x}{T_l - T_u}, \quad (\text{Eq. 3.2})$$

where $0 \leq T_l < T_u < T$ (or T_c) defines the bursty duration and N'_x is the number of documents containing x falling into the time range $[T_l, T_u]$.

That is to say, the word feature x in a news stream is bursty if it appears in a large number of documents over a finite time window, which is distinct from rare words (low document frequency) and stopwords (consistently high document frequency).

However, there are a few challenges we have to tackle in calculating the *burstiness* as elaborated below.

- (i) The *burstiness* is defined with respect to a certain time window. Within T , words often have multiple bursty durations. We have to select the significant bursts for our topic detection task.
- (ii) Some words have periodic bursty durations within T . We need to precisely estimate the burst's periodicity so that each bursty duration $[T_l, T_u]$ can be located accurately.
- (iii) Rare words could also be bursty if they relate to some unimportant (less-reported) topics. We have to differentiate between various strength patterns of burstiness, e.g., heavy bursty or rare bursty, for studying important and unimportant topics, respectively.

In this research, we are only interested in the dominant bursts of the word feature trajectories as they usually correspond to the larger topics of interest to data analysts. In the following section, we apply spectral analysis on a feature trajectory to collectively quantify the dominant burst, periodicity, and bursty strength so that we can quantitatively incorporate the *burstiness* of word features into topic detection.

3.3.3 Spectral Analysis on Feature Trajectory

In this section, we describe how spectral analysis can be applied to every word trajectory in the corpus to uncover periodic and aperiodic bursts of varying strengths.

Fourier Transform

The Fourier transform can decompose any periodic function into a series of harmonically related sinusoids [Mal99]. All linear time-invariant operators of a stationary time series, i.e., the time periods, can be uncovered by the Fourier transform. Given a word feature x we can decompose its offline³ trajectory vector into a sequence of T complex numbers

³For online feature trajectory analysis, we simply replace T with T_c

$[X(1), \dots, X(T)]$ via the Discrete Fourier Transform (DFT) as follows:

$$X(k) = \sum_{t=1}^T y_x(t) e^{-\frac{2\pi i}{T}(k-1)t}, \quad k = 1, 2, \dots, T. \quad (\text{Eq. 3.3})$$

The original time series of a DFT representation can be reconstructed faithfully using a linear combination of complex sinusoids via the Inverse Discrete Fourier transform (IDFT):

$$y_x(t) = \frac{1}{T} \sum_{k=1}^T X(k) e^{\frac{2\pi i}{T}(k-1)t}, \quad t = 1, 2, \dots, T,$$

where the Fourier coefficient $X(k)$ denotes the amplitude of the sinusoid with frequency k/T .

From Fourier Transform to Topic Detection

We first take a closer look at the discrete Fourier transform in Equation Eq. 3.3, whose exponential part can be expanded as follows:

$$e^{-\frac{2\pi i}{T}(k-1)t} = \cos(2\pi \frac{k-1}{T}t) + i \cdot \sin(2\pi \frac{k-1}{T}t).$$

Intuitively, DFT first multiplies the word feature signal $y_x(t)$ with a complex expression comprising sinusoids of frequencies k/T at two different phases (sin and cos). The result is then integrated over the entire time duration T (or T_c for online series). If the integration result is relatively large for some k , then the signal has a major frequency component at k/T and vice versa.

One frequently cited limitation of DFT is that it cannot faithfully reconstruct non-stationary time series such as one-shot bursts, compared to location based approaches like the wavelet transform. In addition, the DFT can only extract all frequency components available over all time, but is unable to pin point the exact time location where each frequency component (aperiodic or periodic burst) occurs. However, this is not a major concern for us since we are not interested in reconstructing the time series; our primary goal is to find the dominant frequency, if any.

The entire news stream is generally⁴ a multi-dimensional non-stationary time series comprising n feature trajectories. By only extracting and using bursty features to reconstruct topics, we are effectively removing the noisy non-bursty features, thereby performing

⁴Except for stopwords, common words, and periodical words.

feature selection on the entire word feature space. Within a bursty feature trajectory, we are further only interested in the dominant frequency, which best portrays the dominant “burst of activities”. All other lesser frequencies/bursts are effectively discarded as noise. The advantage of Fourier transform is more obvious for online trajectories. We can always assume that some interval $(T_c - u, T_c)$ of a non-stationary time series (from 0 to $+\infty$) approaches a stationary time series. In this research, we actually propose using a sliding window to model such a stationary interval, assuming that the dominant frequency, if any, can be detected within a sliding window. In other words, a non-stationary word feature over all time could be become stationary within a finite time window.

Furthermore, unlike the locations of other lesser frequencies, the dominant frequency usually corresponds uniquely to one or multiple bursts over T (or T_c). Therefore, its time location can be easily identified by some simple density-estimation methods, which will be described in details later. In view of all of the above, DFT is actually a very effective tool for analyzing non-stationary feature trajectories.

Dominant Period and Dominant Power Spectrum

The original feature trajectory can be reconstructed with just the dominant frequencies, which can be determined from the power spectrum using the popular periodogram estimator. A periodogram is a sequence of squared magnitude of Fourier coefficients, $\|X(k)\|^2$, $k = 1, 2, \dots, \lceil T/2 \rceil$, which indicates the signal power at frequency k/T in the spectrum. From the power spectrum, the dominant period is chosen as the inverse of the frequency with the highest power spectrum, as follows.

Definition 3.7 (Dominant Period) *The dominant period (DP) of a given feature x is $P_x = T / \arg \max_k \|X(k)\|^2$.*

Accordingly, we have

Definition 3.8 (Dominant Power Spectrum) *The dominant power spectrum (DPS) of a given feature x is*

$$S_x = \|X(k)\|^2, \text{ with } \|X(k)\|^2 \geq \|X(j)\|^2, \forall j \neq k.$$

The periodogram estimator actually provides a straightforward way to quantify the dominant burst of a word feature trajectory: the DPS measures the strength of the dominant

burst while the DP estimates its dominant periodicity. Both DPS and DP are calculated over the entire duration (for online analysis, it is incrementally computed up to the current time t) so that we are able to quantify a *burst* without needing to first find the bursty duration. On the contrary, conventional definitions of Eq. 3.2 typically must compute the bursty duration first. Moreover, based on DPS and DP, word features can be naturally categorized along two dimensions, bursty strength and periodicity, so that each category of word features will correspond to one type of topics. In the next Section, we shall discuss this issue in-depth.

3.4 Identifying Bursty Features from a Historical Stream

In this section, we show how bursty features can be extracted from a historical text stream and used to detect both major and minor topics, which may be periodic or aperiodic.

3.4.1 Categorizing Features based on Power Spectrum and Periodicity

The DPS of a feature trajectory is a strong indicator of its activeness at the specified frequency; the higher the DPS, the more likely for the feature to be bursty.

As a matter of fact, we can combine DPS with DP to identify the set of discriminative features for each type of detected topics (periodic and aperiodic, important and unimportant) by categorizing all features into four types:

- HH: high S_x , aperiodic or long-term periodic ($P_x > \lceil T/2 \rceil$);
- HL: high S_x , short-term periodic ($P_x \leq \lceil T/2 \rceil$);
- LH: low S_x , aperiodic or long-term periodic;
- LL: low S_x , short-term periodic.

The boundary between long-term and short-term periodic is naturally set to $\lceil T/2 \rceil$. However, distinguishing between a high and low DPS is not straightforward, which will be tackled later. In the following section, we shall discuss the relationships between each type of feature set and its corresponding topic detection task, along with the quantitative categorization for different feature categories.

Properties of Different Feature Sets

To better understand the properties of HH, HL, LH and LL, we select four features, *Christmas*, *soccer*, *DBS* and *how* as illustrative examples. Since the boundary between high and low power spectrum is unclear, these chosen examples have a relative wide range of power spectrum values. Figure 3.2.a shows the DFIDF trajectory for *Christmas* with a distinct burst around Christmas day. For the 1-year Reuters dataset, “Christmas” is a typical aperiodic topic, so that its discriminative bursty feature *Christmas* is classified as a HH feature with $S_x = 135.68$ and $P_x = 365$, as shown in Figure 3.2.b. Clearly, the value of $S_x = 135.68$ is reasonable for a well-known bursty topic like Christmas.

Figure 3.2: Feature *Christmas* with relative high S_x and long-term P_x .

The DFIDF trajectory for *soccer* is shown in Figure 3.3.a, from which we see a regular burst every 7 days that is actually confirmed by its computed value of $P_x = 7$, as shown in Figure 3.3.b. Using the domain knowledge that the soccer game has more matches every Saturday, which makes it a typical and heavily reported periodic topic, we thus consider the value of $S_x = 155.13$ to be high and classify *soccer* into the HL set.

Figure 3.3: Feature *soccer* with relative high S_x and short-term P_x .

From the DFIDF trajectory for *DBS* in Figure 3.4.a, we can immediately deduce *DBS* to be an infrequent word in the LH set with a trivial burst on 08/17/1997 corresponding to *DBS Land Raffles Holdings plans*. This is confirmed by the relatively low power of $S_x = 0.3084$ and the long period of $P_x = 365$ as shown in Figure 3.4.b. Moreover, since this aperiodic topic is only reported in a few news articles over a very short span of a few days, we therefore say that its low power value of $S_x = 0.3084$ is representative of unimportant topics.

Figure 3.4: Feature *DBS* with relative low S_x and long-term P_x .

The most confusing example is shown in Figure 3.5 for the word feature *how*, which looks very similar to the graph for *soccer* in Figure 3.3. At first glance, we may be tempted to group both *how* and *soccer* into the same category of HL or LL since both distributions look similar and have the same dominant period of approximately a week. However, further analysis indicates that the periodicity of *how* is due to the differences in document counts for weekdays (averaged 2,919 per day) and weekends⁵ (averaged 479 per day). One would have expected the “periodicity” of a stopword like *how* to be a day. Moreover, despite our DFIDF normalization (see Equation Eq. 3.1, which normalize DF by the number of document per day), the weekday/weekend imbalance is still present; stopwords occur 4 times more frequently on weekends than on weekdays on average. Thus, the DPS remains the only distinguishing factor between *how* ($S_x = 32.13$) and *soccer* ($S_x = 155.13$). However, it is very dangerous to simply conclude that a power value of $S = 32.13$ corresponds to a stopwords feature.

Before introducing our solution to this problem, let’s look at another more typical LL example as shown in Figure 3.6 for *beenb*, which is actually a confirmed typo (the original

⁵“weekends” here also include public holidays falling on weekdays.

Figure 3.5: Feature *how* as an example confusing with feature “soccer”.

sentence is “...any money that had beenb taken”). We therefore classify *beenb* as a noisy feature that does not belong to any topic. Clearly, the trajectory of *how* is very different from *beenb*, which validates our idea that the former has to be considered separately.

Figure 3.6: Feature *beenb* with relative low S_x and short-term P_x .

Stopwords (SW) Feature Set

Based on the above analysis, we realize that there must be another feature set between HL and LL that corresponds to the set of stopwords. Features from this set have moderate DPS and low but known dominant period. Since it is hard to distinguish this feature set from HL and LL only based on DPS, we introduce another factor called average DFIDF ($\overline{DFIDF}$). As shown in Figure 3.6, features like *how* usually have a lower DPS than a HL feature like *soccer*, but have a much higher $\overline{DFIDF}$ than another LL noisy feature such as *beenb*. Since such properties are usually characteristics of stopwords, we group features like *how* into the newly defined stopwords (SW) feature set.

Since setting the DPS and $\overline{DFIDF}$ thresholds for identifying stopwords is more of an art than science, we proposed the HS (Heuristic Stopwords) algorithm, Algorithm 3.1. The

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

basic idea is to only use news stories from weekdays to identify stopwords. The SW set is initially seeded with a small set of 29 popular stopwords utilized by the Google search engine.

Algorithm 3.1 Heuristic Stopwords detection (HS)

Input: Seed SW set, weekday trajectories of all words, feature corpus F

- 1: From the seed set SW, compute the maximum DPS as UDPS, maximum $\overline{DFIDF}$ as UDFIDF, and minimum of $\overline{DFIDF}$ as LDFIDF.
- 2: **for** $x_i \in F$ **do**
- 3: Compute DFT for x_i .
- 4: **if** $S_{x_i} \leq UDPS$ and $\overline{DFIDF}_{x_i} \in [LDFIDF, UDFIDF]$ **then**
- 5: $x_i \rightarrow SW$
- 6: $F = F - x_i$
- 7: **end if**
- 8: **end for**

Overview of Feature Categorization

After the SW set is generated, all stopwords are removed from F . We then set the boundary between high and low DPS to be the upper bound of the SW set's DPS. An overview of all five feature sets is shown in Figure 3.7.

Figure 3.7: The 5 feature sets for topics.

An example for each feature type is also illustrated in Figure 3.7: on the extreme left, three examples from top to bottom are for LH, SW and LL respectively; on the far right, two examples are for HH and HL, respectively. Intuitively only features from HH, HL and

LH are meaningful and could potentially be discriminative for some topics. Therefore, we pruned all feature classified as LL or SW.

3.4.2 Detecting Bursts via Density Models

In this section, we describe how bursts can be identified for features from HH, HL and LH. Unlike Kleinberg’s burst identification algorithm [Kle02], we can identify both significant and trivial bursts without the need to tune any parameters.

Locating Aperiodic Features’ Bursts

For each feature in HH and HL, we truncate its trajectory by keeping only the bursty period, which is modeled with a Gaussian distribution as follows:

$$\mu_x = E(t) = \sum_{t=1}^T y_x(t) \times t,$$

$$\sigma_x = \sqrt{E(t^2) - E^2(t)} = \sqrt{\sum_{t=1}^T y_x(t) \times t^2 - \mu_x^2}.$$

For example, Figure 3.8 shows the word feature *Iraq* with a burst approximately on 09/06/1996 being modeled as a Gaussian (in red). Its bursty period is defined by $[\mu_x - \sigma_x, \mu_x + \sigma_x]$ as shown in Figure 3.8.b.

3.8.a: original DFIDF:time

3.8.b: identifying burst

Figure 3.8: Modeling *Iraq*’s time series as a truncated Gaussian with $\mu = 09/06/1996$ and $\sigma = 6.26$.

Locating Periodic Features’ Bursts

Since we have computed the DP for a periodic feature x , we can easily model its periodic feature trajectory y_x using a mixture of $k = \lfloor T/P_x \rfloor$ Gaussians:

$$f(y_x = y_x(t) | \theta_x) = \sum_{i=1}^k \alpha_i \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2\sigma_i^2}(y_x(t) - \mu_i)^2},$$

where the parameter set $\theta_x = \{\alpha_i, \mu_i, \sigma_i\}_{i=1}^k$ comprises:

- α_i is the probability of assigning y_x into the i^{th} Gaussian. $\alpha_i > 0, \forall i \in [1, k]$ and $\sum_{i=1}^k \alpha_i = 1$;
- μ_i/σ_i is mean/standard deviation of the i^{th} Gaussian.

Thus, the likelihood of feature x is

$$f(y_x|\theta_x) = \prod_{t=1}^T \sum_{i=1}^k \alpha_i f(y_x(t)|\mu_i, \sigma_i).$$

For studying the distribution of feature x for a certain topic, we need to separate the likelihood of x into k parts. At each time point t , we assign x into one of the k Gaussians by

$$L(y_x(t)) = \arg \max_i p_i(t), i = 1, 2, \dots, k,$$

where $L(y_x(t))$ is the Gaussian label for $y_x(t)$ and $p_i(t) = P(L(y_x(t)) = i|y_x(t))$. We subsequently divide the distribution of feature x into k Gaussians as

$$\{f(y_x = y_x(t)|\theta_x)\}_{i=1}^k.$$

Each Gaussian represents one periodic topic, and is modeled similarly as mentioned in Section 3.4.2. The well known Expectation Maximization (EM) [DLR77] algorithm is used to compute the mixing proportions α_i , as well as the individual Gaussian density parameters μ_i and σ_i as follows.

- Expectation step

$$p_i(t) = \frac{\alpha_i f(y_x(t)|\mu_i, \sigma_i)}{\sum_{i=1}^k \alpha_i f(y_x(t)|\mu_i, \sigma_i)},$$

- Maximization step

$$\begin{aligned} \alpha'_i &\leftarrow \frac{1}{T} \sum_{t=1}^T p_i(t), \mu'_i \leftarrow \frac{\sum_{t=1}^T p_i(t) y_x(t)}{\sum_{t=1}^T p_i(t)}, \\ \sigma'_i &\leftarrow \sqrt{\frac{\sum_{t=1}^T p_i(t) (y_x(t) - \mu'_i)^2}{\sum_{t=1}^T p_i(t)}}. \end{aligned}$$

3.5 Reconstructing Offline Topics

After completing our feature trajectory analysis on all features, we should have identified a set of bursty features. The next step is to reconstruct bursty topics from this set of bursty features.

3.5.1 Correlations Among Discriminative Bursty Features

If two bursty features x_i and x_j are discriminative with respect to the same topic, they must satisfy the following necessary conditions:

- (i) x_i and x_j are identically distributed: $y_{x_i} \sim y_{x_j}$.
- (ii) x_i and x_j have a high document overlap.

Measuring Feature Distribution Distance

We measure the distance between two bursty features x_i and x_j using the discrete KL-divergence defined as follows.

Definition 3.9 (Feature Distance) $KL(x_i, x_j)$ is given by $\max(KL(x_i|x_j), KL(x_j|x_i))$, where

$$KL(x_i|x_j) = \sum_{t=1}^T f(y_{x_i}(t)|\theta_{x_i}) \log \frac{f(y_{x_i}(t)|\theta_{x_i})}{f(y_{x_j}(t)|\theta_{x_j})}. \quad (\text{Eq. 3.4})$$

Since KL-divergence is not symmetric, we define the distance between x_i and x_j as the maximum of $KL(x_i|x_j)$ and $KL(x_j|x_i)$. Further, the distance between two aperiodic features can be computed using a closed form of the KL-divergence [Pen01], which is given by

$$KL(x_i|x_j) = 0.5 \log \left(\frac{\sigma_{x_i}^2}{\sigma_{x_j}^2} + \frac{\mu_{x_i}^2 + \mu_{x_j}^2 + \sigma_{x_j}^2 - 2\mu_{x_i}\mu_{x_j}}{2\sigma_{x_i}^2} - 0.5 \right).$$

The same discrete KL-divergence formula of Eq. 3.4 is employed to compute the distance between two periodic features,

Next, we define the overall distance among a set of features $F' \subset F$ using the maximum inter-feature KL-Divergence value as follows.

Definition 3.10 (Set's Distance) $KL(F') = \max_{\forall x_i, x_j \in F'} KL(x_i, x_j)$.

Document Overlap

Let D_i be the set of all documents containing feature x_i . Given two features x_i and x_j , the overlapping document set containing both features is $D_i \cap D_j$. Intuitively, the higher the $|D_i \cap D_j|$, the more likely that x_i and x_j are correlated. We define the degree of document overlap between two features x_i and x_j as follows.

Definition 3.11 (Feature DF Overlap) $d(x_i, x_j) = \frac{|D_i \cap D_j|}{\min(|D_i|, |D_j|)}$.

Likewise, the DF Overlap among a set of features F' is also defined as follows.

Definition 3.12 (Set DF Overlap) $d(F') = \min_{\forall x_i, x_j \in F'} d(x_i, x_j)$.

3.5.2 Unsupervised Greedy Topic Detection

We use bursty features from HH to detect important aperiodic topics, bursty features from LH to detect less-reported/unimportant aperiodic topics, and bursty features from HL to detect periodic topics. All of them share the same algorithm. Given bursty feature $x_i \in HH$, the goal is to find highly correlated bursty features from HH. The set of bursty features similar to x_i can then collectively describe a topic. Specifically, we need to find a subset F'_i of HH that minimizes the following cost function:

$$C(F'_i) = \frac{KL(F'_i)}{d(F'_i) \sum_{x_j \in F'_i} S_{x_j}}, \quad F'_i \subset HH. \quad (\text{Eq. 3.5})$$

The underlying topic c (associated with the burst of x_i) can be represented by F'_i as

$$y(c) = \sum_{x_j \in F'_i} \frac{S_{x_j}}{\sum_{x_u \in F'_i} S_{x_u}} y_{x_j}. \quad (\text{Eq. 3.6})$$

The burst analysis for topic c is exactly the same as the feature trajectory.

The cost in Eq. 3.5 can be minimized using our UG (Unsupervised Greedy) topic detection algorithm, which is used to identify all offline topics and described in Algorithm 3.2. Offline topics are emitted one by one in ascending order of cost by the UG algorithm. In practice, the algorithm can be terminated whenever newly detected events are deemed trivial. The UG algorithm allows a feature to be contained in multiple topics so that we can detect several topics happening at the same time. Furthermore, trivial topics only containing *year/month* features (i.e., a topic only containing 1 feature *Aug* could be identified over a 1-year news stream) could be easily removed, although such topics will have inherent high cost and should already be ranked very low. Note that our UG algorithm only requires one data-dependant parameter, the boundary between high and low power spectrum, to be set once at the onset, and this parameter can be easily estimated using the HS algorithm (Algorithm 3.1).

3.6 From Online Bursty Features to Online Topics

We shall extend the feature trajectory analysis to online bursty feature and online topic detections in this section. In a real-time environment, we will never be able to analyze the “full” corpus and “entire” duration. Therefore, the boundary between high and low power spectrums could initially fluctuate before eventually reaching a steady state. Adjusting

Algorithm 3.2 Unsupervised Greedy topic detection (UG).

Input: bursty features from HH, document index for each bursty feature.

```

1: Sort and select bursty features in descending DPS order:  $S_{x_1} \geq \dots \geq S_{x_{|HH|}}$ .
2:  $k = 0$  and  $C = \emptyset$  for the set of returned topics.
3: for  $x_i \in HH$  do
4:   Init:  $F'_i \leftarrow x_i$ ,  $C(F'_i) = 1/S_{x_i}$ ,  $HH' = HH$  and  $HH' = HH' - x_i$ .
5:   while  $HH'$  not empty do
6:      $m = \arg \min_m C(F'_i \cup x_m)$ .
7:     if  $C(F'_i \cup x_m) < C(F'_i)$  then
8:        $F'_i \leftarrow x_m$  and  $HH' = HH' - x_m$ .
9:     else
10:      break while.
11:    end if
12:  end while
13:  Identify  $c_k$  as Eq. 3.6.
14:  if  $c_k \notin E$  then
15:    Output  $c_k$  and  $C = C \cup c_k$ .
16:     $k = k + 1$ .
17:  else
18:    drop  $c_k$ .
19:  end if
20: end for

```

this boundary incrementally based on the seen data is not recommended because it is actually data-dependent, which makes the DPS categorization criteria inconsistent initially. In the following sections, we thus only focus on analyzing the DP trend for online feature trajectories.

3.6.1 Dominant Period Trend Analysis for Online Bursty Features

We are not able to simply categorize bursty features into fixed classes in the online scenario. Given a long enough period, every feature has an equal chance to exhibit a burst (for example, an Asian name like “been”, if classified into a stopword and discarded, will prevent us from detecting a major topic involving a potentially famous/notorious person named “been”). We first select the same examples (refer to Section 3.4.1), *Christmas*, *soccer*, *DBS*, *how* and *beenb* to illustrate how DP evolves in time. Limited by the 1-year Reuters Corpus, we only analyze the online trends that evolve up to a maximum time of one year.

We know from hindsight that the word *Christmas* from the HH quadrant will exhibit just one major burst. As it ascends the trajectory eventually leading to the maximum burst value, its DP is observed to initially fluctuate, but stabilizes once it reaches the

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

peak, until the end of the period (assuming that the same topic will never happen again). For example, Figure 3.9.b shows that the P_x of *Christmas* stabilizes from the 69th day (10/27/1996) onwards, i.e., at any time k , $P_x = k$ for $365 \geq k \geq 69$. We say that after the 69th day, the P_x trend for *Christmas* becomes linear with time. This is a very important discovery that can help us detect a burst reliably.

Figure 3.9: HH example for the trend of P_x .

Now, let's look at the DP trend of a feature from the HL quadrant, which exhibits regular bursts with a constant period. We discovered that the DP for this feature set actually stabilizes to a constant value after the initial fluctuations. For example, Figure 3.10.b shows that from the 46th day (10/4/1996) onwards, the P_x for *soccer* remains constant: for any time k , $P_x = 7$ for $365 \geq k \geq 46$.

Figure 3.10: HL example for the trend of P_x .

The DP trend of a feature from the LH quadrant behaves similarly as a HH feature like *Christmas*, an example of which is shown in Figure 3.11.b, where the P_x trend for *DBS* converges to a monotonically increasing linear function after the 325th day (7/10/1997).

The features from both the LL and SW quadrants do not contain any meaningful bursts, which can be seen from their haphazard P_x trends that never stabilizes. For example,

Figure 3.11: LH example for the trend of P_x .

Figures 3.12.b and 3.13.b show the DP trends for *beenb* and *how*, respectively, both of which oscillate wildly over time. These observations match our expectation that an accidental feature like *beenb* or an evenly distributed stopword like *how* do not have any consistent periodicities⁶.

Figure 3.12: LL example for the trend of P_x . Figure 3.13: SW example for 2-state of P_x .

⁶Actually, the periodicity for stopwords typically fluctuates around a mean value, which is around 3-7 days for the word *how*, which can be used as a secondary evidence for detecting stopwords online.

Limitations of The Observation Window

The above observations are however limited to the observation window size and data time span. For example, when the DP for the word *Christmas* has stabilized, we simply conclude that $P_x = T_c$ for the current time T_c . This is because the size of the observation window is much smaller than the true bursty period (1 year), which cannot be deduced from our 1-year data.

As far as our algorithm is concerned, events with periodicity larger than 1 year are considered aperiodic, since we only have one year's worth of data. Moreover, the question to ask is whether it is possible to detect periodic events with bursty periods longer than the observation window size but much shorter than 1-year? Figure 3.14 plots the dominant period trend for one such example, the word *Unaudited*, whose bursty period is one quarter or 3 months as shown in Figure 3.1.b.

Figure 3.14: The dominant period trend of the word *Unaudited*.

From this example, the DP trend settled down ($P_x = T_c$) at around 53 days for any observation window that is less than 3 months, since a period of one quarter is relatively aperiodic for such a small window size. However, after 3 months, the detected burst would end. Upon arrival of the second burst, $P_x = T_c/2$, we could then backtrack on our previous decision of treating *Unaudited* as aperiodic and output it as a periodic bursty feature instead, effectively promoting it to state q_2 . Likewise, when the third burst arrives, we would check whether P_x is approximately equal to $T_c/3$, and so on until P_x eventually approaches a constant. With the arrival of more bursts, the periodicity estimate of *Unaudited* would become more and more accurate, i.e., from $216/2 = 113$ to $303/3 = 101$ when the third burst ends, and eventually to $365/4 = 91$.

In other words, we can partially eliminate the effect of the relative small observation window size by continuing to monitor a feature after it has reached the “aperiodic stable” state. In particular, for each word, we continue to monitor all subsequent bursts for changes in its DP until it stabilizes. In the early stages, we might detect a feature as aperiodic bursty, but subsequently promote it to a periodic bursty feature upon seeing more bursts.

Removal of Stopwords

Not all stopwords like *how* can be differentiated from periodic features due to the typical unbalanced nature of the number of documents between weekdays and weekends. The only difference between them remains in their DFIDF values. Therefore, to further remove stopwords, we utilize a similar heuristic training process as we have used for the historical stream (Section 3.4.1). However, now we estimate the range of DFIDF and power spectrum from the stopwords seeds and test all other features based only on the initial sliding window (l days) results.

3.6.2 Modeling The Dominant Period Trend with a 3-state Automata

Based on the insight gained from analyzing the DP trend in Section 3.6.1, we notice that the DP trends for all features can be generalized into 3 types/states: monotonically increasing linear function, constant function, and oscillatory. Referring to Newton’s laws of motion [New87], we define *stable* and *non-stable* states as follows.

Definition 3.13 (Stable State) *If the P_x trend of the online feature trajectory of word feature x is a monotonically increasing linear function ($P_x = t/\alpha$ for each time t , where α is any positive integer constant) or a constant function ($P_x = c$, where c is a constant), we say the P_x trend is in a stable state.*

Definition 3.14 (Non-stable State) *If the P_x trend of the online feature trajectory of word feature x is neither monotonically and linearly increasing nor constant for each time t , we say the P_x trend is in a non-stable state.*

We further divide the *stable* state into two categories, *aperiodic stable* and *periodic stable*, inspired by the analysis on the limitation of the observation window in Section 3.6.1 as follows.

Definition 3.15 (Aperiodic Stable State) *If the P_x trend of the online feature trajectory of word feature x follows $P_x = t$ for each time t , we say the P_x trend is in an aperiodic stable state.*

Definition 3.16 (Periodic Stable State) *If the P_x trend of the online feature trajectory of word feature x follows $P_x = t/\alpha$, where α increases gradually until P_x eventually becomes a constant c , we say the P_x trend is in a periodic stable state.*

We can thus model the P_x trend of each online feature trajectory as a 3-state automaton model A . Figure 3.15 illustrates the three states, q_0 (*non-stable*), q_1 (*aperiodic stable*) and q_2 (*periodic stable*). If the automaton A for a certain feature has remained in state q_1 or

Figure 3.15: 3-state automaton for capturing the dominant period trend.

q_2 for an extended period of time, we qualify it as a bursty feature candidate, which may be periodic or aperiodic. On the other hand, if A remains at state q_0 , we will continue to monitor it for possible future promotion into a stable state. If A remains at state q_1 , we will also continue to monitor it as it could later evolve into a periodic bursty feature.

Such a model can be used to detect online bursty features conveniently. The automaton A for every feature starts in state q_0 . Given the current time T_c (including the first time unit), A has a state sequence $\mathbf{q}(T_c)$, abbreviated as $\mathbf{q}$ and $\mathbf{q} = [q_{i(1)}, \dots, q_{i(T_c)}]$, where $i(t) \in \{0, 1, 2\}, 1 \leq t \leq T_c$. For simplicity, the reverse transitions ($q_1 \rightarrow q_0$, $q_2 \rightarrow q_1$ and $q_2 \rightarrow q_0$) are currently not allowed in A to avoid complications, but could be explored in future work⁷.

For each word, there are two types of state transitions, *burst transition* (from q_0 to q_1 or q_2), and *periodicity transition* (from q_1 to q_2). For *burst transition*, $\mathbf{q}$ has three choice of next states: $\mathbf{q}^0 = [q_0, \dots, q_0]$, $\mathbf{q}^1 = [q_0, \dots, q_0, q_1]$, and $\mathbf{q}^2 = [q_0, \dots, q_0, q_2]$. For *periodicity transition*, $\mathbf{q}$ has two choice of next states: $\mathbf{q}^3 = [q_0, \dots, q_0, q_1, \dots, q_1]$ and $\mathbf{q}^4 =$

⁷Revoking a feature in a periodic stable state could partially alleviate the imbalance of news data between weekdays and weekends, since such an imbalance could possibly lead to a periodic local maxima for a stopword incorrectly classified as a weekly periodic burst.

$[q_0, \dots, q_0, q_1, \dots, q_1, q_2]$. If and when the estimated probability of $\mathbf{q}^1$ or $\mathbf{q}^2$ exceeds that of $\mathbf{q}^0$, we qualify that feature as bursty by adding it to our bursty feature candidate pool at time T_c . After transiting into state q_1 , if at some time, $\mathbf{q}^4 > \mathbf{q}^3$, we subsequently promote it to the third and final periodic state q_2 . When state q_2 has stabilized (e.g., the period approaches a constant) for a certain feature, we stop monitoring it and assume it will remain so forever. Tracking periodicity changes (including termination of a once-periodic event) will be left for future work.

Suppose that until time T_c , the emitted feature trajectory is $y_x(T_c)$, abbreviated as y_x where $y_x = [y_x(1), \dots, y_x(T_c)]$. Applying Bayes theorem, we have

$$P(\mathbf{q}|y_x) = \frac{P(\mathbf{q})P(y_x|\mathbf{q})}{P(y_x)}, \quad (\text{Eq. 3.7})$$

where $P(\mathbf{q}|y_x)$ is used to estimate the probability that A is in state sequence $\mathbf{q}$ at time T_c . $P(\mathbf{q})$ is the *a priori* probability of state $\mathbf{q}$. For non-stationary processes, any distribution assumption could be risky. We therefore model the non-stationary process as a random walk. That is to say, at each time the trajectory has an equal chance to be bursty or non-bursty. Accordingly, $P(\mathbf{q}^0) = P(\mathbf{q}^1) = P(\mathbf{q}^2) = (1/3)^{T_c}$ for *burst transition* and $P(\mathbf{q}^3) = P(\mathbf{q}^4) = (0.5)^{T_c}$ for *periodicity transition*. Since $P(y_x)$ is also independent of $\mathbf{q}$, Equation Eq. 3.7 can be further simplified as the likelihood of y_x as follows:

$$P(\mathbf{q}|y_x) \propto P(y_x|\mathbf{q}) = \prod_{t=1}^{T_c} P(y_x(t)|q_{i(t)}). \quad (\text{Eq. 3.8})$$

Accordingly, we have

$$\begin{aligned} P(\mathbf{q}^0|y_x) &\propto P(y_x|\mathbf{q}^0) = \prod_{t=1}^{T_c} P(y_x(t)|q_0), \\ P(\mathbf{q}^1|y_x) &\propto P(y_x|\mathbf{q}^1) = P(y_x(T_c)|q_1) \cdot \prod_{t=1}^{T_c-1} P(y_x(t)|q_0), \\ P(\mathbf{q}^2|y_x) &\propto P(y_x|\mathbf{q}^2) = P(y_x(T_c)|q_2) \cdot \prod_{t=1}^{T_c-1} P(y_x(t)|q_0), \\ P(\mathbf{q}^3|y_x) &\propto P(y_x|\mathbf{q}^3) = \prod_{t=T'}^{T_c} P(y_x(t)|q_1) \cdot \prod_{t=0}^{T'-1} P(y_x(t)|q_0), \\ P(\mathbf{q}^4|y_x) &\propto P(y_x|\mathbf{q}^4) = P(y_x(T_c)|q_2) \cdot \prod_{t=T'}^{T_c-1} P(y_x(t)|q_1) \cdot \prod_{t=0}^{T'-1} P(y_x(t)|q_0), \end{aligned}$$

where $P(y_x(t)|q_0)$, $P(y_x(t)|q_1)$ and $P(y_x(t)|q_2)$ are the likelihoods of data in state q_0 , q_1 and q_2 at time t , respectively.

3.6.3 Maximum Likelihood Bursty Feature Detection

There is no need to estimate the whole data likelihood accurately as for both *burst transition* and *periodicity transition*, only the probability comparison at the latest time point is needed. To recap, at current time T_c , a feature in state q_0 will be promoted to state q_1 (bursty aperiodic) whenever $P(y_x(T_c)|q_1)$ ($P(y_x(T_c)|q_2)$) exceeds $P(y_x(T_c)|q_0)$; a bursty aperiodic feature (state q_1) will be further promoted to state q_2 (bursty periodic) whenever $P(y_x(T_c)|q_2) > P(y_x(T_c)|q_1)$. We estimate the likelihood of a data point based on the following two intuitive observations.

Effect of Sliding Window Size

Observation 3.1 *When A enters the stable state, the P_x trend should have at least remained consistent (either linear or constant) for the past l time units.*

This observation can be explained by the example in Figure 3.16, which shows that a feature like *Christmas* may initially exhibit a piece-wise linear DP trend (zone A), i.e., $P_x = t$ for some time t . However, once the word picks up momentum, it's P_x will begin to oscillate.

Figure 3.16: Effect of sliding window on the *stable* state.

In fact, retrospective analysis suggests that the P_x for *Christmas* oscillates irregularly at the initial stage (zone A). This is a normal phenomenon for non-stationary processes. However, we would be more confident of qualifying the *stable* state if the P_x trend remained stable for l consecutive time points. For example, in Figure 3.16 the P_x trends in both zone B and zone C have a much higher probability of qualifying the *stable* state compared to points in zone A. For news streams, due to the unbalanced document frequency among weekdays and weekends as previously discussed in Section 3.4.1, l can be conveniently set to 7 days. That is to say, once a P_x trend remains in the true *stable* state for seven days, it is considered final.

The sliding window size actually assumes some interval of the otherwise non-stationary trajectory to be stationary, which is possible when the sliding window falls into the global bursty period of the aperiodic feature, or if the true period of the periodic feature has been identified. For periodic features, setting a universal sliding window size can be tricky: those periodic features with period much longer than l are possibly identified as aperiodic features. However, this is an open problem for online prediction. First, it is impossible to choose a sliding window to match all unseen periods. Second, an overly large sliding window defeats the very purpose of online prediction. Fortunately, this problem can be partially alleviated by the *periodicity transition* of our 3-state automata model.

Observation 3.1 is also applicable to the other two categories of bursty features. Figure 3.17 superimposes a sliding window of size $l = 7$ on both *soccer* and *DBS* DP trend curves after they have stabilized. We see that their DPs are indeed stationary within the sliding window size after it stabilized, but not prior to that.

3.17.a: soccer(P:day)

3.17.b: DBS(P:day)

Figure 3.17: Effect of a sliding window of size $l = 7$ on other features.

Gradient of Trajectory

Unfortunately, a sliding window of size l alone is not a sufficient condition to confirm a *stable* state. For example, consider zone B in Figure 3.16, A has reached at the *stable* state based on Observation 3.1. However, the original feature trajectory of *Christmas* shown in Figure 3.18 shows that the absolute DFIDE values in zone B to be very small, i.e., it does not vary much compared to points in zone C.

Our domain knowledge on Christmas verifies that A is in fact premature in zone B (before day 98, 11/25/1996). The reason why Observation 3.1 alone is limited is due to the fact that documents corresponding to important non-periodic bursty topics like Christmas are typically spread over a long time range before and after the time of the dominant

Figure 3.18: Effect of the DFIDF trajectory gradient on the *stable* state.

burst. We notice that compared to zone B, zone C (between day 98 and 128, 11/25/1996-12/25/1997) showed much more significant increase in DFIDF values and is thus a better estimate of the true bursty period for the feature *Christmas*. We thus combine the change in DFIDF value with the *stable* state of A in the second observation as follows.

Observation 3.2 Suppose that A enters the stable state at time T_c and its P_x trend has remained consistent for the past $n \geq l$ days, then compared to all previous n days, $DFIDF_x(T_c)$ should show the largest variation (change in gradient), otherwise a promotion to the next state is not allowed.

Figure 3.19 verifies the effectiveness of Observation 3.2 on a periodic feature like *soccer* and a less-important aperiodic feature like *DBS*. In Figure 3.19, the dotted vertical line indicates the time from which the DP trend remains consistent. Comparing with all points after the dotted line, we see that both the periodic and global maxima exhibit maximum variation in DFIDF gradient values. Specifically, Figure 3.19.b further illustrates how Observation 3.1 and Observation 3.2 can be combined to avoid local maxima.

3.19.a: soccer(DFIDF:day)

3.19.b: DBS(DFIDF:day)

Figure 3.19: Largest change in DFIDF trajectory gradient helps to avoid local maxima.

Final Likelihood Estimation

Based on the above two observations, for *burst transition*, we summarize the estimation of the automaton state likelihoods, $P(y_x(t)|q_0)$, $P(y_x(t)|q_1)$ and $P(y_x(t)|q_2)$, as follows ($\bar{q}_0$ is

annotated as either q_1 or q_2).

$$P(y_x(t)|\bar{q}_0) = \begin{cases} 1 & \text{if 1) } P_x(u) \text{ moves regularly, } u \in [t', t] \text{ and } t - t' \geq l, \\ & 2) \Delta DFIDF_x(t) > \Delta DFIDF_x(u), \\ & \text{where } \Delta DFIDF_x(u) = \frac{DFIDF_x(u) - \frac{1}{u} \sum_{k=1}^u DFIDF_x(k)}{\frac{1}{u} \sum_{k=1}^u DFIDF_x(k)}, \\ & 3) \text{ and } DFIDF_x(t) > \frac{1}{t} \sum_{k=1}^t DFIDF_x(k); \\ 0 & \text{otherwise.} \end{cases}$$

$$P(y_x(t)|q_0) = 1 - P(y_x(t)|\bar{q}_0).$$

We use the gradient of $DFIDF_x$ over a moving average to measure its amount of change at time t . The first condition originates from Observation 3.1, while the second condition stems from Observation 3.2. An additional third condition is needed to constrain the burst in the rise to the peak.

Upon detection of a bursty feature, we categorize it into aperiodic or periodic as follows.

(i) Aperiodic ($P(y_x(t)|q_1) = 1, P(y_x(t)|q_2) = 0$), if

$$P_x(u) = u, u \in [t - l, t].$$

(ii) Periodic ($P(y_x(t)|q_1) = 0, P(y_x(t)|q_2) = 1$), if

$$P_x(u) = P_x(t - l), u \in [t - l + 1, t].$$

Similarly, for *periodicity transition*, a word will remain in state q_1 with $P(y_x(t)|q_1) = 1$ until the following conditions are satisfied.

$$P(y_x(t)|q_2) = \begin{cases} 1 & \text{if 1) } P_x(u) = \frac{u}{\alpha}, u \in [t', t], t - t' \geq l \text{ and } \alpha = 2, 3, \dots, \\ & 2) \text{ conditions 2 and 3 in Eq. 3.9;} \\ 0 & \text{otherwise.} \end{cases}$$

In practice, once α increases to 2 from 1, we could announce the *periodicity transition* from state q_1 to state q_2 . However, it is interesting to trace the whole periodicity evolvement until the period converges to a constant, which is the true period. During this convergence process, α is incremented by 1 each time until u/α equals the true period.

3.6.4 Predicting Online Topics

The process of uncovering online topics is the same as Algorithm 3.2, except that we run it every day, where all detected bursty features are recorded along with their dominant power spectrums. We can use the first-order average DFIDF scores to determine the actual time interval for aperiodic bursty features. For example, if at some time t , the DFIDF score for a bursty word falls below the moving average of its previous DFIDF scores, we can decide that the burst has died at time t . For periodic bursty features, we simply provide the corresponding period.

3.7 Performance Evaluation

In this section, we study the performances of our bursty feature detection methods and topic detection algorithms. We first introduce the dataset and experimental setup for all experiments. Second, we evaluate the categorization of features for HH, HL, LH, LL, and SW for the historical text stream, followed by the subjective study on the (a)periodic topic detection problem with Algorithm 3.2. Then, we examine the results of online bursty features detection. The evaluations on online topic detection are separated into two parts: (1) quantitative benchmark of our online topic detection algorithms with the baseline approaches, using top offline topics as ground truth, and (2) subjective examination of the detected online (a)periodic topics.

3.7.1 Dataset and Experimental Setup

The Reuters Corpus contains 806,791 English news stories from 08/20/1996 to 08/19/1997 time-stamped at the day resolution. Version 2 of the open source Lucene software [luc] was used to tokenize the news text content and generate the document-word vector. In order to preserve the time-sensitive past/present/future tenses of verbs and the differences between lower case nouns and upper case named entities, no stemming was done. Since dynamic stopwords removal is one of the functionalities of our method, stopwords were not removed. We did remove non-English characters, however, after which the number of word features amounts to 423,433. All experiments were implemented in Java and conducted on a 3.2 GHz Pentium 4 machine running Windows 2003 Server with 1 GB of memory.

3.7.2 Retrospectively Categorizing Features

We downloaded a set of 34 well-known stopwords utilized by the Google search engine as our seed training features, which include *a, about, an, are, as, at, be, by, de, for, from, how, in, is, it, of, on, or, that, the, this, to, was, what, when, where, who, will, with, la, com, und, en* and *www*. We excluded the last five stopwords as they are uncommon in news stories. By only analyzing news stories over 259 weekdays, we computed the upper bound of the power spectrum for stopwords at 11.18 and corresponding $\overline{DFIDF}$ ranges from 0.1182 to 0.3691. Any feature f satisfying $S_x \leq 11.18$ and $0.1182 \leq \overline{DFIDF}_x \leq 0.3691$ over weekdays will be considered a stopwords. In this manner, 470 stopwords were found and removed as shown in Figure 3.20.

Figure 3.20: Distribution of SW (stopwords) in the HH, HL, LH, and LL regions.

Some detected stopwords include *A* ($P = 65$, $S = 3.36$, $\overline{DFIDF} = 0.3103$), *At* ($P = 259$, $S = 1.86$, $\overline{DFIDF} = 0.1551$), *GMT* ($P = 130$, $S = 6.16$, $\overline{DFIDF} = 0.1628$) and *much* ($P = 22$, $S = 0.80$, $\overline{DFIDF} = 0.1865$). After the removal of these stopwords, the distribution of weekday and weekend news are more or less matched, and in the ensuing experiments, we shall make use of the full corpus (weekdays and weekends).

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

The upper bound power spectrum value of 11.18 for training stopwords was selected as the boundary between the high power and low power spectrums. The boundary between high and low periodicity was set to $\lceil 365/2 \rceil = 183$. All 422,963 (423,433 – 470) word features were categorized into 4 feature sets: HH (69 features), HL (1,087 features), LH (83,471 features), and LL (338,806 features) as shown in Figure 3.21.

Figure 3.21: Distribution of categorized features over the four quadrants (shading in log scale).

In Figure 3.21, each gray level denotes the relative density of features in a square region, measured by $\log_{10}(1+n_k)$, where n_k is the number of features within the k -th square region. From the figure, we can make the following observations:

- (i) Most features have low S and are easily distinguishable from those features having a much higher S , which allows us to detect important (a)periodic topics from trivial topics by selecting features with high S .
- (ii) Features in the HH and LH quadrants are aperiodic, which are nicely separated (big horizontal gap) from the periodic features. This allows aperiodic and periodic topics to be identified subsequently without any conflicts.

- (iii) The (vertical) boundary between high and low power spectrum is not as clear cut and the exact value will be data specific.

By checking the scatter distribution of features from SW on HH, HL, LH, and LL as shown in Figure 3.20, we found that 87.02%(409/470) of the detected stopwords originated from LL. The LL classification and high $\overline{DFIDF}$ scores of stopwords agree with the generally accepted notion that stopwords are equally frequent over all time. Therefore, setting the boundary between high and low power spectrum using the upper bound S_x of SW is a reasonable heuristic.

3.7.3 Retrospectively Detecting Aperiodic Topics

We shall evaluate our two hypotheses, 1) important aperiodic topics can be defined by a set of HH features, and 2) less reported aperiodic topics can be defined by a set of LH features. Since no benchmark news streams exist for topic detection (TDT datasets are not real streams), we evaluate the quality of the automatically detected topics in comparison to manually-confirmed topics found by searching through the corpus.

Among the 69 HH features, we detected 17 important aperiodic topics as shown in Table 3.1 ($c_1 - c_{17}$). Note that the entire identification took less than 1 second, after removing topics containing only the *month* feature.

Among the 17 topics, other than the overlaps between c_3 and c_4 (both belongs to the same hostage topic), c_{11} and c_{16} (both on company reports), the 14 identified topics are surprisingly accurate and correspond very well to the major topics of the period. For example, the defeat of Bob Dole, election of Tony Blair, Missile attack on Iraq, etc. Recall that selecting the features for one topic should minimize the cost in Eq. 3.5 such that 1) the number of features spans different topics, and 2) not all features relevant to a topic will be selected, e.g., the feature *Clinton* is discriminative to c_{12} but since *Clinton* relates to many other topics, its time domain signal is far different from those of other discriminative features like *Dole* and *Bob*. The number of documents of a detected topic is roughly estimated by the number of indexed documents containing all discriminative features. We can see that all the 17 detected aperiodic topics are in fact popularly reported topics of that time period.

After 742 minutes of computation time, we detected 23,525 less reported aperiodic topics from 83,471 LH features. Table 3.2 lists the top 5 detected aperiodic topics ($c_{18} - c_{22}$) with respect to the cost. We found that these 5 topics are actually very trivial topics with only

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

Table 3.1: All important aperiodic topics ($c_1 - c_{17}$).

ID	Discriminative Bursty Features	Bursty Pe- riod	# Doc	True Topic
c_1	Sali, Berisha, Albania, Alba- nian, March	02/02/1997- 05/29/1997	1409	Albanian's president Sali Berisha lost in an early election and resigned, 12/1996-07/1997.
c_2	Seko, Mobutu, Sese, Kabila	03/22/1997- 06/09/1997	2273	Zaire's president Mobutu Sese coordinated the native rebellion and failed on 05/16/1997.
c_3	Marxist, Peru- vian	11/19/1996- 03/05/1997	824	Peru rebels (Tupac Amaru Revolutionary Movement) led a hostage siege in Lima in early 1997.
c_4	Movement,Tupac, Amaru,Lima, hostage,hostages	11/16/1996- 03/20/1997	824	The same as c_3 .
c_5	Kinshasa,Kabila, Laurent,Congo	03/26/1997- 06/15/1997	1378	Zaire was renamed the Democratic Republic of Congo on 05/16/1997.
c_6	Jospin, Lionel, June	05/10/1997- 07/09/1997	605	Lionel Jospin was appointed Prime Minister on 06/02/1997.
c_7	Iraq, missile	08/31/1996- 09/13/1996	1262	U.S. fired missile at Iraq on 09/03/1996 and 09/04/1996.
c_8	Kurdish, Bagh- dad,Iraqi	08/29/1996- 09/09/1996	1132	Iraqi troop fought with Kurdish faction around 09/1996.
c_9	May, Blair	03/24/1997- 07/04/1997	1049	Tony Blair became the Primary Minister of the UK on 05/02/1997.
c_{10}	slalom, skiing	12/05/1996- 03/21/1997	253	Slalom Game of Alpine Skiing in 01/1997-02/1997.
c_{11}	Interim, months	09/24/1996- 12/31/1996	3063	Tokyo released company interim results for the past several months in 09/1996-12/1996.
c_{12}	Dole, Bob	09/09/1996- 11/24/1996	1599	Bob Dole lost the 1996 US presidential election.
c_{13}	July, Sen	06/13/1997- 08/18/1997	344	Cambodia's Prime Minister Hun Sen launched a bloody military coup in 07/1997.
c_{14}	Hebron	10/15/1996- 02/14/1997	2098	Hebron was divided into two sectors in early 1997.
c_{15}	April, Easter	02/23/1997- 05/04/1997	480	Easter feasts around 04/1997 (for western and Orthodox).
c_{16}	Diluted, Group	04/27/1997- 07/20/1997	1888	Tokyo released all 96/97 group results in 04/1997-07/1997.
c_{17}	December, Christmas	11/17/1996- 01/26/1997	1326	Christmas feast in late 12/1997.

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

a few news reports, and are usually subsumed by some larger topics. For example, c_{22} is one of the rescue topics in an airplane hijack topic. One advantage of our UG Algorithm for discovering less-reported aperiodic topics is that we are able to precisely detect the true topic period.

Table 3.2: Top 5 less-reported aperiodic topics ($c_{18} - c_{22}$).

ID	Discriminative Features	Bursty Period	# Doc	True Topic
c_{18}	Kolaceva,winter,Together, promenades, Zajedno, Slobodan, Belgrade, Serbian, Serbia, Draskovic, municipal, Kragujevac	1/25/1997	3	University students organized a vigil on Kolaceva street against government on 1/25/1997.
c_{19}	Tutsi,Luvengi,Burundi, Uvira,fuel,Banyamulenge, Burundian,Kivu,Kiliba,Runingo, Kagunga,Bwegera	10/19/1996	6	Fresh fighting erupted around Uvira between Zaire armed forces and Banyamulengs Tutsi rebels on 10/19/1996.
c_{20}	Malantacchi,Korea,Guy, Rider,Unions,labour,Trade, unions,Confederation, rammed,Geneva,stoppages, Virgin,hire,Myongdong, Metalworkers	1/11/1997	2	Marcello Malantacchi secretary general of the International Metalworkers Federation and Guy Rider who heads the Geneva office of the International Confederation of Free Trade Unions attacked the new labour law of South Korea on 1/11/1997.
c_{21}	DBS, Raffles	8/17/1997	9	The list of the unit of Singapore DBS Land Raffles Holdings plans on 8/17/1997.
c_{22}	preserver, fuel, Galawa, Huddle, Leul, Beausse	11/24/1996	3	Rescued a woman and her baby during a hijacked Ethiopian plane that ran out of fuel and crashed into the sea near Le Galawa beach on 11/24/1996.

3.7.4 Retrospectively Detecting Periodic Topics

Among the 1,087 HL features, 330 important periodic topics were detected within 10 minutes of computing time. Table 3.3 lists the top 5 detected periodic topics with respect to the cost ($c_{23} - c_{27}$). All of the detected periodic topics are indeed valid, and correspond to real life periodic topics. The GMM model is able to detect and estimate the bursty period nicely

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

although it cannot distinguish the slight difference between every Monday-Friday and all weekdays as shown in c_{23} . We also notice that c_{26} is actually a subset of c_{27} (soccer game), which is acceptable since the Sheffield league results are announced independently every weekend.

Table 3.3: Top 5 important periodic topics ($c_{23} - c_{27}$).

ID	Discriminative Bursty Features	Bursty Period	# Doc	True Topic
c_{23}	PRICE,LISTING,MLN, MATURITY,COUPON,MOODY, AMT,FIRST,ISS,TYPE, PAY,BORROWER	Monday-Friday/week	7966	Announce bond price on all weekdays.
c_{24}	Unaudited, Ended, Months, Weighted, Provision, Cost, Selling, Revenues, Loss, Income, except, Shrs, Revs	every season	2264	Net income-loss reports released by companies in every season.
c_{25}	rating, Wall, Street, Ian	Monday-Friday/week	21767	Stock reports from Wall Street on all weekdays.
c_{26}	Sheffield, league, scoring, goals, striker, games	Friday-Sunday/week	574	Match results of Sheffield soccer league were published on Friday, Saturday and Sunday 10 times than other 4 days.
c_{27}	soccer, matches, Results, season, game, Cup, match, victory, beat, played, play, division	Friday-Sunday/week	2396	Soccer games held on Friday, Saturday and Sunday 7 times than other 4 days.

3.7.5 Detecting Online Bursty Features

After a running time of 9 minutes 18 seconds, we identified and removed 776 stopwords based only on the initial sliding window (set $l = 7$) data. The process is based on Algorithm 3.1, with the upper bound of 0.5711 for DPS, the upper bound of 0.3977 for $\overline{DFIDF}$ and the lower bound of 0.0950 for $\overline{DFIDF}$. About 98.98% of the 470 offline stopwords (ground truth) are accounted in the 776 discovered online stopwords, which indicates the high recall of Algorithm 3.1 for online processing.

Altogether, we identified 129,169 online aperiodic bursty features and 36,677 online periodic bursty features. Figure 3.22 plots the distribution of the number of online bursty features found per day. Note that the initial sliding window is not supposed to contain any bursty features. We see that 13,365 bursty features were identified on the first valid day (day 12, 8/31/1996). This unusually large amount is mainly due to existence of local maxima. After several days, the distribution for most words began to stabilize.

Figure 3.23 further verifies the effect of local maxima by categorizing the DPS and DP respectively. From Figure 3.23.a, we see that the major portion (72.67%) of detected bursty features have DPS less than 0.001. The effect of the sliding window size (l) is illustrated in Figure 3.23.b. Bursty features with DP equal to or less than l are preferred over those with DP higher than l , because many periodic bursty features with DP higher than l are categorized as aperiodic. Despite the above limitations, which are inevitable for online processing, we find that the online bursty features detected by us do have a clear range of quantifiable DPS. By specifying any user-preferred threshold, highly ranked bursty features with high DPS can be easily put into practice for tasks such as reconstructing online topics.

3.7.6 Superiority of the Online Bursty Feature Detection Method

It is not straightforward to even subjectively evaluate the superiority of our online bursty feature detection method, simply because there exists no benchmark dataset. In this section, we compare our online bursty feature detection method with other standard techniques, treating the top offline topics ($c_1 - c_{27}$) as the ground truth (actually 26 distinct topics, since c_3 and c_4 are duplicated). Although the collection of $c_1 - c_{27}$ might not match all real-life topics exactly, it still provides a fair reference for the various techniques.

Figure 3.22: Distribution of online bursty features ($\#$ detected bursty features:day).

Figure 3.23: Categorization of detected online bursty features.

The baseline methods include GRADIENT and STD. In GRADIENT, the feature x is identified as a bursty feature at time t , if condition 2 and condition 3 are satisfied (refer to Section 3.6.3). In STD, the feature x is identified as a bursty feature at time t , if

$$DFIDF_x(t) > \mu_x(\tau) + \sigma_x(\tau), \quad (\text{Eq. 3.9})$$

where $\mu_x(\tau)$ and $\sigma_x(\tau)$ are the mean and standard deviation of DFIDF trajectory until time t , respectively.

The two methods proposed by us are denoted as DP+GRADIENT and DP+STD, with the latter combining the conditions of STD and condition 1 of DP+GRADIENT. The evaluation measures are defined as follows.

- (i) *Accurate alarm*: at least one discriminative bursty feature of a ground truth topic is identified and located in its bursty period; for periodic ground truth topics, we check the first bursty period.

- (ii) *Premature alarm/delayed alarm*: all detected discriminative bursty features occur before/after the bursty period; for periodic ground truth topics, we compare with the first bursty period.
- (iii) *False alarm*: periodic ground truth topics are announced as aperiodic topics, and vice-versa; or there is a complete mismatch of bursty periods.
- (iv) *Miss*: None of a topic's discriminative bursty features can be identified;

We say an online bursty feature detection method is better if it detects a larger number of accurate alarms, premature alarms, and delayed alarms, but fewer false alarms and misses, in this preferred order. Intuitively, we would prefer accurate alarms over premature alarms, followed by delayed alarms and false alarms. Misses are not desirable.

Table 3.7.6 shows the comparison results. We see that both GRADIENT and STD suffer from heavy misses, which are unacceptable for a practical topic alert system. Our methods, DP+GRADIENT and DP+STD largely reduced the number of misses. Generally, DP+GRADIENT achieves the best performance and is thus used in this research. The result verifies that the 3-state automaton model of DP is able to help track online trends better.

Table 3.4: Online topic detection benchmarks.

Method\Alarm type:	Accurate	Premature	Delayed	False	Miss
GRADIENT	3	10	1	1	11
STD	5	4	1	0	16
DP+GRADIENT	4	11	8	3	0
DP+STD	3	12	6	4	1

3.7.7 Detecting Online Aperiodic Topics

We reconstructed 73,432 online aperiodic topics based on 129,169 aperiodic bursty features. The distribution of the number of detected topics per day is shown in Figure 3.24. Note that our algorithm does not check for duplicated topics, which will linearly increase the computational time as more topics are detected. Thus, a major topic might be detected multiple times since its various discriminative bursty features are possibly identified at different times. However, as the discriminative bursty features changes, we could say that the same topic is actually induced from different sub-topics. Figure 3.25 presents the cost

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

categorization for all online aperiodic topics, from which we see that the majority of online aperiodic bursty features with low DPS largely belongs to online aperiodic topics with extremely high cost (more than 1000).

Figure 3.24: Distribution of online aperiodic topics (# detected topics:day).

Figure 3.25: Cost categorization of detected online aperiodic topics.

As a practical application for the second subjective evaluation phase, we find all online aperiodic topics with minimal cost per 30 days in Figure 3.26, which are detailedly listed as ($c_{28} - c_{39}$) in Table 3.5. All of them are true aperiodic events (only *burst transitions* were triggered).

Among the 12 topics, the alarms of topics c_{35} and c_{36} are delayed, but the other 10 topics correspond very well to the lifespans of real-life topics. Again, the number of documents for each topic is roughly estimated as the number of indexed documents containing the discriminative features during the bursty period. In fact, the fewer the number of documents, the higher the chance with which the topic will be delayed. The reason is that we can only realistically confirm a spike-like topic after a certain time duration. We can see that all 12 aperiodic topics are accurate (not false alarms). On the other hand, c_{30} and c_{31} refers to the same topic, so do c_{32} and c_{33} . However, they are represented by different bursty

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

Table 3.5: The online aperiodic topics ($c_{28} - c_{39}$) with minimal cost per 30 days.

ID	Discriminative Bursty Features	Detected Date	# Doc	True Topic
c_{28}	Fran, Kitty, Devil, Manteo, Nags, skating, overwash	9/08/96	58	Hurricane Fran. 8/26/1996-9/17/1996.
c_{29}	Saddam, Baghdad, Patriotic, PUK, KDP, Jalal, Ta- labani, Barzani, Massoud, Jemjal	10/14/96	90	Kurdish factional fighting in northern Iraq between Kurdistan Democratic Party KDP (leadered by Massoud Barzani) and the Patriotic Union of Kurdistan PUK (leadered by Jalal Talabani). 8/23/1996-11/15/1996.
c_{30}	specified, Parent, Hunet	11/16/96	5	Tokyo released the interim results of the past six months for Hunet Inc. 11/07/1996-11/28/1996.
c_{31}	ACTUAL, Kensetsu	11/23/96	13	Tokyo released the interim results of the past six months for Dai-ichi Kensetsu, Dia Kensetsu Co Ltd, Nippon Dentsu Kensetsu Co Ltd and Saeki Kensetsu Kogyo Co Ltd. 11/14/1996-11/22/1996.
c_{32}	Lima, Tudela	1/12/97	56	Peruvian rebels led a four-month hostage in Lima. The hostages include the Peruvian Foreign Minister Francisco Tudela. 12/18/1996-3/29/1997.
c_{33}	Marxist, Velazco	2/15/97	102	Isaac Velazco, a Hamburg-based self-styled international spokesman, reported the hostage led by Marxist MRTA rebels. 12/24/1996-4/28/1997.
c_{34}	schemes, Lushnje	3/02/97	55	The clash between Albanians fearing the loss of their savings in collapsing pyramid schemes and the government in the southern town of Lushnje. 1/23/1997-4/17/1997.
c_{35}	skiing, Filischkin	4/09/97	15	Results related to Russian athlete Andrei Filischkin in Alpine skiing world championships. 12/16/1996-3/09/1997.
c_{36}	Dutch, anatum	4/20/97	9	Dutch food group Nutricia faced the product recall because of a suggested link with the rare Salmonella anatum infection. 1/24/1997-2/10/1997.
c_{37}	Taleban, Dostum	5/25/97	112	The Taleban mounted a summer campaign against the forces of General Abdul Rashid Dostum who controlled northern Afghanistan. 5/19/1997-6/24/1997.
c_{38}	Sassou, Nguesso, Kolelas, Hellot, Mampouya	7/06/97	14	A bloody showdown between Congo President Pascal Lissouba and former Marxist military leader Denis Sassou Nguesso. Reuters updated the reports from Kolelas's spokesman Hellot Mampouya. 6/10/1997-7/25/1997.
c_{39}	TIME, Trust	7/27/97	1134	Bombay banks quoted the forward premia on dollar. 8/19/1996-8/20/1997, but bursted on 5/1997-8/1997.

Figure 3.26: Online aperiodic topics with minimal cost per 30 days (topic cost:day).

features, most likely each representing a sub-topic of the same topic. For example, both c_{32} and c_{33} are related to the topic “the hostage crisis in Lima”, but c_{32} mainly focus on the fact that Peruvian Foreign Minister Francisco Tudela are among the held hostages, and c_{33} refers to the commentary of Isaac Velazco, who is a Hamburg-based self-styled international spokesman, on this topic. Such *duplicate* topics validated our explanation about the large amount of detected aperiodic sub-topics, which can be meaningful.

Another encouraging result is that many of the 12 detected topics actually relate to the important offline topics ($c_1 - c_{17}$) listed in Table 3.1. For example, c_{29} originates from c_8 , c_{30} and c_{31} are sub-topics of c_{11} , c_{32} and c_{33} are sub-topics of c_3 , c_{34} comes from c_1 , and c_{35} belongs to c_{10} , etc. In other words, our online detection algorithm was able to locate many global maxima, especially for those bursty features with low cost. The same result also validated the practicality of using a low cost threshold to detect important online topics.

3.7.8 Detecting Online Periodic Topics

We identified 34,786 online periodic topics based on 36,677 periodic bursty features. The distribution of the number of detected topics per day is shown in Figure 3.27. Apart from sharing the same explanation as aperiodic topics on the large amount of detected topics (duplicated topics, small local maxima, etc.), another reason why most of periodic topics only contain one 1 feature is that we require bursty periodic features to have strictly matching periods. For example, “soccer game” is a typical periodic topic with a period of 7 days. However, only “soccer” was identified as a periodic feature with a 7-day period,

while the period of “game” is very different, thus these two features cannot be grouped together. Figure 3.28 shows the cost categorization for all online periodic topics. We see that compared to aperiodic topics, the portion of periodic topics with high cost (more than 1000) is largely reduced (from 73.35% to 27.73%), because far fewer local maxima occur periodically. Again, the clear demarkation of topic costs help us locate the most obvious detected topics every day.

Figure 3.27: Distribution of online periodic topics (# detected topics:day).

Figure 3.28: Categorizations for detected online periodic topics.

In the following we subjectively evaluate some online periodic topics. It is pointless to track the periodic topics with the lowest cost per 30 days again since some typical periodic topics will dominate the evaluation. Instead, we manually examine the corresponding online result of the top 5 ground truth periodic topics $c_{23} - c_{27}$ obtained via offline detection. Our method (DP+GRADIENT) achieved 4 delayed alarms and 1 false alarm in total, as shown in Figure 3.29-Figure 3.32. More specifically, c_{24} triggered the *burst transition* in the first period, and subsequently triggered the *periodicity transition* in the second period so that the false alarm became a delayed alarm afterwards. The other topics only triggered the *burst transition*.

CHAPTER 3. DISCRIMINATIVE FEATURES OF NEWS STREAMS AND WORD-LEVEL TOPIC DETECTION

Figure 3.29: Online detection status versus offline ground truth for c_{23} . Bars denote periods. Delayed alarm.

Figure 3.30: Refined online detection status versus historical ground truth for c_{24} . Bars denote periods. False alarm in the first period and delayed alarm starting from the second period.

Figure 3.31: Online detection status versus offline ground truth for c_{25} . Bars denote periods. Delayed alarm.

In fact, in most cases it is almost impossible to accurately detect the first burst of a periodic topic on the fly, due to the constraint of the sliding window. In other words, the

Figure 3.32: Online detection status versus offline ground truth for c_{26} and c_{27} . Bars denote periods. Delayed alarm.

dominant period of a periodic topic will only remain stable after a few periods. For example, c_{23} is delayed by 8 periods, c_{25} is delayed by 13 periods and both c_{26} and c_{27} are delayed by 7 periods. The topic occurring periods detected online will not always match that of the offline ground truth exactly as well. For example, initially online c_{23} can only be detected on Wednesday, after two more periods, its coverage extends to Monday-Thursday, missing Friday when the offline ground truth still occurs. Similar observations also happened on c_{25} , c_{26} and c_{27} .

We take a closer look at c_{24} , where a false alarm was produced in the first period. Figure 3.30 indicates that c_{24} was falsely detected as an aperiodic topic, whose occurring period roughly matches the first burst of the periodic topic c_{24} . We know for a fact that this is because the size of the sliding window is much smaller than the true bursty period (one quarter or 3 months) of c_{24} . However, with our 3-state automata, after the first “false alarm” (at state q_1 , where *burst transition* occurred when the first burst came), we could achieve three delayed alarms (at state q_2 , where *periodicity transition* occurred when the second burst came) subsequently. Moreover, if we do not care about the periodicity of an event and only output the event itself (bursty periods and representative bursty features), there is no false alarm at all.

3.8 Summary

This chapter took a whole new perspective of analyzing feature trajectories as time domain signals. By considering the word document frequencies in both time and frequency domains, we were able to derive many new characteristics about news streams that were previously

unknown, e.g., the different distributions of stopwords during weekdays and weekends, the periodicity of topics, etc. For the first time in the area of TDT, we applied a systematic approach to automatically detect important and less-reported, periodic and aperiodic topics for both offline and online news streams.

The key idea of our work lies in the observations that (a)periodic topics have (a)periodic discriminative features and (un)important topics have (in)active discriminative features, differentiated by their power spectrums and time periods. To address the offline topic detection, a simple and effective mixture density-based approach was used to identify feature bursts and their associated bursty periods. For online topic detection, we proposed a robust and principled approach based on the evolution of each feature trajectory's dominant period. Each feature's dominant period can be in one of three possible ordered states: *non-stable*, *aperiodic stable*, and *periodic stable*. A feature that has transitioned into either of the two stable states is considered bursty. Once a set of bursty features have been confirmed, we then applied our unsupervised greedy algorithms to detect aperiodic and periodic topics in both batch and online modes. Simulation on the real-world Reuters news stream showed very promising results for both offline and online topic detection.

In the next chapter, we shall consider another approach, the document level approach for topic detection.

Chapter 4

Temporal Discriminative Probabilistic Models for Document-Level Topic Detection

Despite the significant progress we have made on word-level topic detection in Chapter 3, the classical document-level topic detection approach remains still useful for certain applications that deals mostly with documents. In this chapter, we first systematically examine several probabilistic and non-probabilistic temporal models for document-level topic detection. We then propose a Discriminative Probabilistic Model (DIS), which makes effective use of the set of temporal discriminative (bursty) features that was described in Chapter 3. Our DIS model was benchmarked against several existing models and found to yield much better practical topic detection performance.

4.1 Motivation for Temporal Discriminative Probabilistic Models

From a data-mining perspective, online and offline document-level TD may seem no different from incremental and offline document clustering, respectively. However, there are a number of subtle TD characteristics, which if not taken into due consideration, can adversely affect practical clustering performances: 1) time plays a pivotal role, with every news document bearing a time stamp, 2) news topics are naturally bursty, i.e., new topics are constantly generated while old topics die off, 3) news documents with semantically similar content but disparate time-frames most likely originated from different topics, e.g., hurricane Mitch of October 1998 and hurricane George of September 1998 are two distinct

topics that share a large amount of vocabulary. TD can thus be viewed as a special case of stream clustering [GMM⁺03], with the clustering portfolio at any time point akin to a concept that drifts with time [Tsy04, HCL06b].

Despite the fact that time plays an important role, the vast majority of existing TD solutions [YPC98, ALMS00, SC01, YZCJ02, BCF03, KA04] do not explicitly incorporate time into their formulations; each news document is represented as a vector with time-agnostic static weights, with just one minor procedural modification: news vectors are processed in time-stamp order, i.e., online TD is used to handle the temporal factor. The entire setup smacks of ad hocism that builds upon the TFIDF (term frequency inverse document frequency) vector space model [SB88], which itself is lacking in mathematical formalism¹. Fortunately, this simple procedural modification over static text representation model seems to work quite well in practice.

The question to ask is then this: can a concise text representation model be formulated to explicitly capture the temporal element in news streams, and yet remain practical and effective for both online and offline TD? In this chapter, we seek to answer this question by evaluating several time-aware probabilistic formulations for TD, and proposing our own temporal and discriminative probabilistic framework called DIS.

DIS is a Bayesian probabilistic framework that considers each news document as a point in discriminative word *and* time vector space. The posterior topic probability given a document *and* time is computed and subsequently used to assign a document to multiple fitting topics. One unique feature of DIS is that it only needs to operate on a subset of topic and temporal discriminative words instead of the full vocabulary space, making DIS very efficient for practical implementation.

Disregarding the time component for now, DIS actually belongs to a more general class of probabilistic mixture frameworks that ranges from Bayesian models to mixture generative models like mixture of Gaussians [Das99] or von-Mises Fisher (vMF) [BDGS03] distributions, and finally to full Bayesian generative ones like Latent Dirichlet Allocation (LDA) [BNJ03] hierarchical topic models. Table 4.1 positions DIS among the different probabilistic topic models. From Table 4.1, we can see that probabilistic models like DIS are simple extensions of deterministic models; it estimates the posterior probability of a

¹The reformulation of TFIDF in a probabilistic framework by Joachims [Joa97] does however help improve the standing of TFIDF.

Table 4.1: Probabilistic TD models.

Model	Document	Topic	TD approach
Deterministic	point	point	k -means
Discriminative Probabilistic	point	probability	DIS
Mixture	point	distribution	mixture of {vMF, Gaussians}
Mixed Membership	distribution	distribution	LDA

topic given observed documents from the class-conditional document probabilities. Going one step further, mixture generative models treat each topic as a distribution over points (documents). Mixed membership generative models such as LDA treat both documents and topics as distributions; documents are distributions of topics, topics are distributions of words, and words independently originate from documents.

Why did we choose to incorporate time into a simple probabilistic model instead of the more sophisticated generative models? In fact, we have tried adding the temporal element to mixture generative models like vMF, but that did not result in any significant improvements in TD performance. On the other hand, mixed membership generative models like LDA have already been extended to handle time [BL06]. In general, we found that simple probabilistic models seem to achieve the best balance in terms of model complexity and performance, validating the principal of Occam’s Razor [Ari76].

Formulating DIS using a probabilistic framework makes it more amenable to statistical analysis and practical deployment. For example, using a similar approach to Joachims [Joa97], we show that DIS is equivalent to a temporal version of TFIDF with discriminative feature selection, which can be trivially implemented. Lastly, soft topic assignment is a natural and practical application of DIS that can be enabled at will, and useful for TD in practice. In contrast, deterministic approaches like k -means hard-clustering assign all documents into a set of disjoint clusters, where each cluster represents one topic.

4.2 Document-Level Topic Detection Model

Document-level TD aims to group thematically-related documents from a temporal text stream into an unknown number of topics. The working definition adopted throughout this chapter assumes that each document could be assigned to multiple topics, and no topic is known beforehand. Formally, let D be a news stream starting at time t_0 , with

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

varying number of documents $N(t)$ published at each discrete time point $t \geq t_0$ and total number of documents $N = \sum_t N(t)$ in D . The ordering of documents at different times are maintained, but does not matter for documents published at the same discrete time point. Let Z be a set of gold-standard topics for text stream D . Furthermore, suppose there exists a ground-truth concept $G : D \times Z \rightarrow [0, 1]$ which is the posterior probability distribution of $P(z|\mathbf{d})$: assigning a document vector variable $\mathbf{d}$ to a class label variable z into the close real interval $[0, 1]$. Here, we shall only consider unsupervised learning, which is more practical for online topic detection. As such, $G(\mathbf{d}, z)$ is never used for training, but only referenced in the final evaluation as ground truth. Thus, our goal is to learn a topic detection model H , which is formally defined as follows.

Definition 4.1 (Topic Detection Model) *Through unsupervised learning, we would like to find a hypothesis distribution*

$$H : D \times C \rightarrow [0, 1], \quad (\text{Eq. 4.1})$$

to approximate the ground truth mapping G as closely as possible. $H(\mathbf{d}, c)$ is a probability distribution for $P(c|\mathbf{d})$, governing how a document d is assigned to each cluster c . The objective is to find a hypothesis that maximizes accuracy, i.e., the overlap between the two mappings H and G .

We further break down the topic detection problem into two sub-problems: *offline topic detection* and *online topic detection*.

Definition 4.2 (Offline Topic Detection) *The offline topic detection model learns a hypothesis $H : D \times C \rightarrow [0, 1]$ at one go using all documents throughout the text stream D for all time $t \geq t_0$. The objective is to recover the underlying true class labels Z for the complete corpus D . $P(c|\mathbf{d})$ denotes the offline topic distribution of any document d .*

Definition 4.3 (Online Topic Detection) *The online topic detection model learns a hypothesis $H : D \times C \rightarrow [0, 1]$ using only documents up to the time stamp of the most recent incoming document d . The objective is a point assignment that identifies cluster c for document d at time t , denoted as $\arg \max_c P(c|\mathbf{d}, t)$. Note that for online detection, cluster assignment is irrevocable once assigned.*

Online topic detection is modeled as a point assignment process rather than a distribution because the total number of topics/clusters is unknown and actually increases over time. In other words, at any time point, the distribution space is incomplete.

Table 4.2 further summarizes the differences between offline and online topic models.

Table 4.2: Comparisons between offline/batch and online topic models.

Type	Document ordering	Applications	# topics k	Topic assignment
offline	Irrelevant for all t	Post analysis of topics	Fixed	Distribution
online	Relevant for different t	Identify new topics in real time	Increasing	Point

4.3 Topic Detection Models

Topic detection models can be broadly classified into two types: non-probabilistic and probabilistic. Non-probabilistic topic detection models based on hard clustering have achieved decent topic detection performances in the past, and will be described in Section 4.3.1. On the other hand, probabilistic models have never been specifically applied to topic detection. In sections 4.3.2 and 4.3.3, we shall introduce two popular probabilistic models, vMF mixture and LDA, respectively, along with ways in which they can be applied to topic detection.

4.3.1 Non-probabilistic Models

Non-probabilistic models cluster documents directly, which are modeled as vectors in high-dimensional word space, where the relationship between documents and words are explicitly linear and independent [SB88]. We picked the spherical k -means (SPK) clustering as the representative non-probabilistic topic detection model.

SPK on Offline Topic Detection

Suppose there are N document vectors $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_N$ in $\mathbb{R}^n$ from data stream D , and each document vector has been normalized to lie on the unit hypersphere, i.e., $\|\mathbf{d}_i\|=1 \forall i$. The popular TFIDF [SB88] vector space model is used to generate each document vector as

$$\begin{aligned} \mathbf{d}_i &= [w_{1i}, w_{2i}, \dots, w_{ni}]^T, \\ w_{ji} &= f_{ji} \log(1 + \frac{N}{N_j}), \end{aligned} \quad (\text{Eq. 4.2})$$

where f_{ji} is the term frequency of word x_j in document d_i and N_j is document frequency of word x_j , i.e., number of documents in which word x_j appears. SPK seeks a partitioning of D into k disjoint clusters $c_1, c_2, \dots, c_k$ that maximizes the following objective function

$$\sum_{j=1}^k \sum_{d \in c_j} \mathbf{d}^T \mathbf{c}_j, \quad (\text{Eq. 4.3})$$

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

where $\mathbf{c}_j$ is the centroid of cluster c_j and defined as

$$\mathbf{c}_j = \frac{\sum_{d \in c_j} \mathbf{d}}{N(c_j)}, \quad (\text{Eq. 4.4})$$

and $N(c_j)$ is the number of documents in cluster c_j . The objective function of Eq. 4.3 measures the overall “coherence” of a clustering assignment. It assumes that a cluster is good if it is cohesive. If the generated clusters are on average good, the resulting topic distribution hypothesis H should ideally approximate the ground-truth mapping G for $k = |Z|$.

However, finding the optimal solution to the above objective function is NP-complete. SPK thus uses a k -means type of approximation algorithm, which is an iterative procedure given below:

- (i) Start with an initial hypothesis H with an arbitrary partitioning of the document vectors. Compute the cluster centroids accordingly.
- (ii) Update the new hypothesis H' , assign each document vector $\mathbf{d}$ to the nearest cluster (centroid),

$$P(c|\mathbf{d}) = \begin{cases} 1 & \text{if } c = \arg \max_{c_j} \mathbf{d}^T \mathbf{c}_j; \\ 0 & \text{otherwise.} \end{cases}$$

- (iii) Update the cluster centroids based on the new topic distribution H' .
- (iv) If the difference between the objective function of H' and H is less than a threshold, H' is output as the solution. Otherwise, set $H = H'$, and go to step 2.

Dhillon and Modha [DM01] have proved that the above iterative process results in a monotonically increasing objective function, which eventually converges to a local maxima. Thus, SPK provides an effective and efficient approximation for maximizing the objective function. The topic assignment $P(c|\mathbf{d})$ relies completely on the cosine similarity between each document and a cluster centroid. However, the clustering does not explicitly compute the posterior probability $P(c|\mathbf{d})$. During each iteration, H is always a hard partition on D . In fact, SPK can be extended to soft-clustering via vMF, which will be discussed in Section 4.3.2.

SPK for Online Topic Detection

One of the most common non-probabilistic models for online topic detection in TDT is the single-link and single-pass clustering [AHF⁺05]. Each document is represented as a bag-of-words vector based on cumulative TFIDF, which is given by

$$w_{x,d} = \frac{f_{x,d} \cdot \log((0.5 + N)/N_x(\tau))}{\log(1.0 + N(\tau))}, \quad (\text{Eq. 4.5})$$

where τ means up to time $t \leq \tau$, $N_x(\tau)$ is the number of documents up to time τ in which word x appears, and $N(\tau)$ is the total number of documents seen to date. For each incoming document d at time t , compute its cosine similarity to every previous document ($\leq t$) in the collection. If its similarity to the nearest neighbor (1-NN) is above a threshold ϵ , assign d to the nearest cluster; otherwise, a new singleton cluster containing d is created. We shall call this non-probabilistic model *Single-Link-All*. Single-Link-All does not estimate the online topic distribution $P(c|\mathbf{d}, t)$ but instead learn it via point estimation.

In fact, short of any domain knowledge such as the prior probability of generating a new topic, the threshold-based process defined in Single-Link-All is the most efficient and effective way for discovering new topics in an unsupervised manner. Throughout the chapter, we will use it as our topic discovery process.

We propose a simple and effective hybrid online-offline strategy for online topic detection, which borrows the idea from k -means clustering models [AHWY03]. A single-pass clustering approach like Single-Link-All typically creates too many small pseudo-topic clusters, as well as a few large clusters containing many irrelevant documents. By periodically applying global clustering on all data seen to date, better clusters can be obtained. The frequency of global clustering can be adjusted based on some quality measures such as desired clustering quality, which will be discussed in greater details later in the experimental part (see Section 4.5.4). The main difference between Single-Link-All and our approach is that we compare each incoming news document to all cluster/topic centroids instead of all news vectors, which makes our approach less susceptible to noise and outliers.

4.3.2 Mixture Generative Models

The vMF mixture model [BDGS03] is a mixture generative probabilistic model that has worked quite well for document clustering. More specifically, vMF mixture estimation can be viewed as a generalized version of SPK clustering. vMF mixtures assume that each

topic/cluster is represented by a von-Mises Fisher distribution over all member documents, and the entire set of document distribution is a mixture of k vMF distributions. Compared to SPK, vMF mixture 1) computes the true topic assignment probability $P(c_j|\mathbf{d})$ of each document d for every topic/cluster c_j , and 2) estimates the distribution of each document d as a mixture of cluster-conditional vMF distributions $P(\mathbf{d}|c_j)$, thereby maximizing the overall likelihood. In other words, SPK only considers the nearest neighbors in its iteration/update, i.e., only the winning cluster is updated, whereas vMF mixtures take into consideration all cluster conditioned distributions.

vMF Mixture on Offline Topic Detection

Given a unit-length document vector $\mathbf{d} \in \mathbb{R}^n$ and $\|\mathbf{d}\| = 1$, we assume that $\mathbf{d}$ is generated by an n -variate vMF distribution with the following probability density function (pdf),

$$f(\mathbf{d}|c, \kappa) = R_n(\kappa) e^{\kappa \mathbf{d}^T \mathbf{c}}, \quad (\text{Eq. 4.6})$$

where $\mathbf{c}$ plays the role of the mean vector of a latent topic, $\|\mathbf{c}\| = 1$, $\kappa > 0$, and $R_n(\kappa)$ is a normalizing constant

$$R_n(\kappa) = \frac{\kappa^{n/2-1}}{(2\pi)^{n/2} I_{n/2-1}(\kappa)}, \quad (\text{Eq. 4.7})$$

where $I_r(\cdot)$ represents the modified Bessel function of the first kind and order r . Clearly, the probability of $\mathbf{d}$ given c will be high if they are more similar to one another, as shown in Eq. 4.6. The concentration parameter κ is sort of like the classical variance parameter in Gaussian distributions, since it modulates the similarity between document $\mathbf{d}$ and topic centroid vector $\mathbf{c}$. In the limit where $\kappa \rightarrow 0$, the cluster conditional pdf for any document $\mathbf{d}$ reduces to a constant, i.e., it degenerates into a uniform distribution over the unit hypersphere. If $\kappa \rightarrow \infty$, all documents will have effectively zero pdf values except for the mean vector, which has infinite pdf.

In the vMF mixture models, a document vector $\mathbf{d}$ is modeled by a mixture of k vMF distributions as

$$f(\mathbf{d}|\Theta) = \sum_{j=1}^k \alpha_j f_j(\mathbf{d}|\theta_j), \quad (\text{Eq. 4.8})$$

where $\alpha_j > 0$ is the topic prior and $\sum_{j=1}^k \alpha_j = 1$, $\theta_j = (c_j, \kappa_j)$, and $\Theta = \{\alpha_1, \dots, \alpha_k, \theta_1, \dots, \theta_k\}$.

Given a data stream D , assume that each document sample d_i is sampled i.i.d. (independently and identically distributed) following the mixture distribution of Eq. 4.8, we can use the Expectation Minimization (EM) algorithm [DLR77] to estimate the parameters of the vMF mixtures. The M-step updates the parameters as follows:

$$\begin{aligned}\alpha_j &= \frac{1}{N} \sum_{i=1}^N P(c_j | \mathbf{d}_i, \Theta), \\ \mathbf{r}_j &= \sum_{i=1}^N \mathbf{d}_i P(c_j | \mathbf{d}_i, \Theta), \\ \bar{\mathbf{r}}_j &= \frac{\|\mathbf{r}_j\|}{\sum_{i=1}^N P(c_j | \mathbf{d}_i)}, \\ \mathbf{c}_j &= \frac{\mathbf{r}_j}{\|\mathbf{r}_j\|}, \\ \kappa_j &= \frac{\bar{\mathbf{r}}_j \cdot n - \bar{\mathbf{r}}_j^3}{1 - \bar{\mathbf{r}}_j^2}.\end{aligned}$$

And the E-step is given by

$$P(c_j | \mathbf{d}_i, \Theta) = \frac{\alpha_j f_j(\mathbf{d}_i | \Theta)}{\sum_{l=1}^k \alpha_l f_l(\mathbf{d}_i | \Theta)}.$$

The above EM algorithm thus defines an iterative soft-clustering process that is guaranteed to converge. Eq. 4.9 computes the individual probability of assigning document d_j to cluster (topic) c_j , which depends on two factors: 1) the topic prior, i.e., larger clusters generally have a higher affinity compared to smaller ones; 2) the cosine similarity between document and cluster centroid (mean direction of vMF distributions). Based on Eq. 4.9, the complete topic distribution can be learned for offline topic detection.

vMF Mixture for Online Topic Detection

An online extension of vMF mixture is proposed in [BB07]. The basic idea is to incrementally update each cluster centroid as new documents are added while keeping the cluster concentration parameter κ and mixing proportions unchanged, as shown below,

$$\mathbf{c}^{(t+1)} = \mathbf{c}^{(t)} + \frac{1}{t+1}(\mathbf{d} - \mathbf{c}^{(t)}), \quad (\text{Eq. 4.9})$$

where $\mathbf{d}$ is the new document vector, $\mathbf{c}^{(t)}$ and $\mathbf{c}^{(t+1)}$ are centroid vectors of the cluster to which $\mathbf{d}$ is assigned before and after the arrival of document d . A trade-off, which

accounts for the effective memory length, is used in practice to increase the influence of a new document [BB07].

Since we assume online topic detection to be a point assignment process, the new topic discovery procedure for both vMF mixtures and Single-Link-All is similar. The only difference between them is that vMF mixture compares a new document with all existing vMF topic distributions, while Single-Link-All uses the 1-NN as the topic of reference.

Although we can easily compute the generative probability of a new document d given an existing vMF distribution, but in the case when d results in a new topic, that new topic will only contain one document d . This means that the generative probability of d given this new topic is practically 1 since we are using d to generate itself. We compute the posterior cluster probabilities given a document in Eq. 4.6 to estimate a document's similarity to every known cluster. If the maximum posterior probability is below a threshold ϵ , a new cluster centered at document d is created. The concentration parameter of the new cluster is initialized to the average κ of all existing clusters.

Naturally, such an incremental clustering scheme deteriorates with increasing number of incoming documents, and global optimization is needed from time to time. Therefore, the vMF mixture is periodically recomputed on all documents seen to date, to ensure that the distribution parameters can be globally optimized.

4.3.3 Mixed Membership Generative Models

Mixed membership generative models represent documents as mixtures of topics, where a topic is a probabilistic distribution over words. In other words, the latent topic variable act as an intermediary between words and documents. This is different from the classic vector space model, which explicitly models a document as a linear combination of orthogonal word vectors. Thus, a document is expressed as a distribution over topics; for each word in that document, a topic is sampled according to the topic distribution, and a word is drawn from every topic with a given probability. In our approach, LDA [BNJ03] is used to invert this process by inferring the set of topics responsible for generating a collection of documents.

LDA on Offline Topic Detection

Latent Dirichlet Allocation (LDA), which is very similar to Probability Latent Semantic Indexing (pLSI), is a static generative topic model that represents each document as a

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

mixture of topics. The only difference between the two is that the latter assumes a Dirichlet topic prior for each document. That is to say, under a uniform Dirichlet prior distribution in which all documents share the same fixed topic prior vector, LDA is equivalent to pLSI. Therefore, LDA can be seen as a more sophisticated version of pLSI. Note that, in general, the number of topics k for LDA must be pre-specified, making it more suitable for offline topic detection.

More formally, in LDA the generative process for each document vector $\mathbf{d}$ is described as below.

- (i) Let the document length be drawn from a distribution, e.g., Poisson or Gaussian. This step is rather trivial, as the choice of distribution can be determined from the corpus data.
- (ii) Estimate the topic prior vector α from the corpus. Based on α , choose the topic prior vector θ for each document vector $\mathbf{d}$ from a Dirichlet distribution² as $\theta \sim \text{Dirichlet}(\alpha)$.
- (iii) For each word x_i in $\mathbf{d}$,
 - (a) Choose a topic c_j , whose topic vector follows the Multinomial distribution $c_j \sim \text{Multinomial}(\theta)$.
 - (b) x_i is generated using the multinomial probability $P(x_i|c_j, \beta)$ where β is an $n \times k$ matrix with $\beta_{ij} = P(x_i|c_j)$.

Mathematically, the above process of generating a document can be summarized as follows:

$$P(\theta, c_j, \mathbf{d}|\alpha, \beta) = P(\theta|\alpha) \prod_{i=1}^n P(c_j|\theta) P(x_i|c_j, \beta). \quad (\text{Eq. 4.10})$$

There are $k + n \times k$ parameters to be estimated: k parameters for α and $n \times k$ parameters for the β matrix (k topics β_j).

The primary goal of the topic model hypothesis is to learn the posterior probabilities of topics given $\mathbf{d}$, which is in turn the inference of LDA, as follows:

$$P(\theta, c_j|\mathbf{d}, \alpha, \beta) = \frac{P(\theta, c_j, \mathbf{d}|\alpha, \beta)}{P(\mathbf{d}|\alpha, \beta)}.$$

The above inference is intractable, and approximation techniques are needed to learn the hypothesis in practice. Such approximations are beyond the scope of our research, and has been previously elaborated in [BNJ03].

²Topics are assumed to follow a multinomial distribution in LDA, and its prior vector α also follows a distribution over documents. Generally, such a distribution is naturally simulated by the multinomial's conjugate prior, which is simply the Dirichlet distribution in Bayesian statistics.

LDA on Online Topic Detection

There is a simple online extension of LDA, known as the Dynamic Topic Model (DTM) [BL06], which assumes that the topic model evolves gradually with time and is distributed normally. More specifically, DTM conducts k -component LDA analysis at each time slice t sequentially, and conditionally defines the natural parameters³ of each topic $\beta_j(t+1)$ to be a Gaussian distribution centered upon the previous value $\beta_j(t+1)|\beta_j(t) \sim N(\beta_j(t), \sigma^2 I)$, where $\beta_{ij}(t) = P(x_i|c_j, t)$ and I is the identity matrix.

However, DTM fixes the number of clusters k for the entire duration of the stream, and does not detect new topics. Therefore, it is not suitable for online topic detection. Zhang et. al. [ZGY05] first attempted to apply LDA on online topic detection via a supervised approach; by using the topic prior of a training dataset to estimate the likelihood of generating new topics.

We adopt a similar threshold-based new topic discovery strategy as the Single-Link-All for LDA. In the online phase, we update the topic model parameters following the above DTM process for all existing topics, and create new topics just as in the vMF mixture approach. The similarity between a document and existing topics is modeled as the generative probabilities defined in Eq. 4.10. If the maximum probability falls below a user defined threshold ϵ , a new topic is created. The parameters of the new topic, $\beta_{ij} = P(x_i|c_j)$, are initialized to the ratio of the term frequency of x_i in the new document over the new document's length. The new topic prior is initialized to the mean prior of all existing topics. Finally, we periodically apply offline LDA to the entire dataset as the global optimization phase.

4.4 Discriminative Probabilistic Models

In this section, we shall propose our discriminative probabilistic models (DIS) for both offline and online topic detection. There are a number of driving factors that led to the formulation of the DIS model. First, existing probabilistic models, especially LDA, seem to be overly complex for the problem of topic detection. Second, although non-probabilistic models

³Due to the nice statistical characteristics of the exponential family, statisticians often prefer rewriting other distributions like Multinomial as another exponential distribution. The parameters of the original Multinomial distribution are called mean parameters, and the parameters of the transferred exponential distribution are called natural parameters.

have worked fairly well for topic detection, up till now there have been no corresponding mathematical explanation. For example, the TFIDF weighting formula (Eq. 4.5) adopted by Single-Link-All is tuned empirically, without any theoretical basis or mathematical insight.

We therefore come up with the simple discriminative (DIS) models for topic detection, with two goals: 1) to simplify the overly complicated probabilistic models by removing the distribution assumption, and 2) to provide a compelling theoretical framework to support non-probabilistic models like the TFIDF. In fact, we will show in later sections that the posterior topic probability given document is a variation on the classical TFIDF formulation for both offline and online topic detection, on the condition that a set of discriminative words can be found.

4.4.1 Offline Discriminative Model

We assume that there exists a feature set $X \in F$ in the text collection D , with which all documents can be topically discriminated from on another with minimal error, where F is the full word/vocabulary feature space. Any feature $x \in X$ can be a discriminative feature for a certain latent topic. For example, for the topic “Hurricane Mitch” in D , the words *wind*, *hurricane*, *Mitch* and *storm* might be proper discriminative features. However, stopwords like *the* will not contribute to any latent topic and are thus treated as non-discriminative features. We use non-discriminative features instead of a predefined set of stopwords because given a data D , there might be other word features that are not among the commonly accepted set of stopwords in information retrieval but yet irrelevant to any latent topic in D . The set of discriminative features is therefore data-dependent.

Given a new document vector $\mathbf{d}$ and an existing topic c_j , we have the following discriminative process:

$$\begin{aligned} P(c_j|\mathbf{d}) &= \sum_{x \in F} P(c_j|x, \mathbf{d})P(x|\mathbf{d}) = \sum_{x \in X} P(c_j|x, \mathbf{d})P(x|\mathbf{d}) + \sum_{x \in F \setminus X} P(c_j|x, \mathbf{d})P(x|\mathbf{d}) \\ &= \sum_{x \in X} \frac{P(\mathbf{d}|c_j, x)}{P(\mathbf{d}|x)} \cdot P(c_j|x)P(x|\mathbf{d}) + \sum_{x \in F \setminus X} P(c_j|\mathbf{d}) \cdot \frac{f_{x,d}}{|\mathbf{d}|}, \end{aligned}$$

where $P(x|\mathbf{d})$ is estimated as $f_{x,d}/|\mathbf{d}|$ assuming only unigram word features are considered⁴. Apparently, $f_{x,d}/|\mathbf{d}|$ is more or less a constant for non-discriminative features, including stopwords, as their term frequencies given a document only depend on the document length

⁴The model can be extended to n-gram word features or other forms of feature representations.

$|d|$. For those non-discriminative features that do not contribute to any latent topic, we can also assume that $P(c_j|x, \mathbf{d}) = P(c_j|\mathbf{d})$. By introducing the constant $R = \sum_{x \in F \setminus X} f_{x,d}/|d|$, we have

$$(1 - R)P(c_j|\mathbf{d}) = \sum_{x \in X} \frac{P(\mathbf{d}|c_j, x)}{P(\mathbf{d}|x)} \cdot P(c_j|x)P(x|\mathbf{d}).$$

Since the discriminative feature (to be called discriminator henceforth) x provides enough information about document d to the extent that no additional information about document d is gained by considering its class label c_j , we can say $P(\mathbf{d}|c_j, x) = P(\mathbf{d}|x)$. We thus have

$$P(c_j|\mathbf{d}) = \frac{1}{1 - R} \sum_{x \in X} P(c_j|x)P(x|\mathbf{d}).$$

Accordingly, the point assignment (assigning the most likely topic) of document d is given by $\arg \max_{c_j} P(c_j|\mathbf{d})$, and its topic distribution $P(c|\mathbf{d})$ is learned via one additional normalization step over all topics as

$$P(c|\mathbf{d}) = \frac{1}{R_C} \sum_{x \in X} P(c|x)P(x|\mathbf{d}), \quad (\text{Eq. 4.11})$$

so that $\sum_{j=1}^k P(c_j|\mathbf{d}) = 1$, where R_C is the normalization factor.

Estimation of Offline Discriminative Model

In the work of [Joa97], the right part of Eq. 4.11 has been shown to be equivalent to the Rocchio classifier by allowing for reasonable variations on the popular TFIDF document representation. We reapply this seminal result onto a clustering framework as follows.

The probability distribution defined in Eq. 4.11 can be rewritten as

$$P(c_j|\mathbf{d}) = \frac{1}{R_C} \sum_{x \in X} P(c_j|x)P(x|\mathbf{d}) = \frac{1}{R_C} \sum_{x \in X} \frac{P(x|c_j)P(c_j)}{\sum_{c_l \in C} P(x|c_l)P(c_l)} \cdot P(x|\mathbf{d}),$$

where $P(x|\mathbf{d})$ is estimated as $f_{x,d}/|d|$, $P(c_j)$ is estimated as $N(c_j)/N$ and $P(x|c_j)$ is estimated as $1/N(c_j) \cdot \sum_{d' \in c_j} P(x|\mathbf{d}')$. Accordingly, $P(c_j|\mathbf{d})$ is reformulated as

$$P(c_j|\mathbf{d}) = \frac{1}{R_C} \sum_{x \in X} \frac{\frac{1}{N} \sum_{d' \in c_j} \frac{f_{x,d'}}{|d'|}}{\sum_{c_l \in C} \frac{1}{N} \sum_{d'' \in c_l} \frac{f_{x,d''}}{|d''|}} \cdot \frac{f_{x,d}}{|d|}.$$

By defining the *term frequency* and the *inverse document frequency* of x as

$$TF'(x, d) = \frac{f_{x,d}}{|d|},$$

$$IDF'(x) = \sqrt{\frac{N}{\sum_{d \in D} \frac{f_{x,d}}{|d|}}},$$

$P(c_j|\mathbf{d})$ is further reformulated as

$$\begin{aligned} P(c_j|\mathbf{d}) &= \frac{1}{R_C} \cdot \frac{N(c_j)}{N} \sum_{x \in X} \left[\frac{1}{N(c_j)} \sum_{d' \in c_j} TF'(x, d') \cdot IDF'(x) \right] \cdot [TF'(x, d) \cdot IDF'(x)] \\ &= \frac{1}{R_C} \cdot \frac{N(c_j)}{N} \cdot \mathbf{d}^T \mathbf{c}_j, \end{aligned} \quad (\text{Eq. 4.12})$$

where

$$\mathbf{c}_j = \frac{1}{N(c_j)} \cdot \sum_{d' \in c_j} \mathbf{d}',$$

and each dimension of a document vector is given by

$$w_i = TF'(x_i, d) \cdot IDF'(x_i).$$

From Eq. 4.12, after embedding documents into the discriminative feature space X , we see that the topic probability distribution of a document is proportional to both the absolute size of the topic and the distance (inner product) from the document to the topic's centroid. This process is equivalent to each iteration of the k -component centroid-based soft clustering based on the TFIDF vector space model, except that in our case, a larger cluster has a higher affinity compared to a smaller one.

Application to Offline Topic Detection

An iterative process is needed to locally optimize the clustering procedure of the discriminative model. For hard partitioning, the SPK algorithm can be used. During each iteration, point assignment of each document is used to determine the cluster label. For soft partitioning, vMF can be used, where the topic probability distribution in Eq. 4.11 is used to assign documents, and the clusters are updated as a mixture of all documents weighted by the posterior cluster probability $P(c_j|\mathbf{d})$.

4.4.2 Online Discriminative Model

In this section, we shall extend the static discriminative model to a dynamic version, where documents at different times are *not exchangeable*. In an online model, both topics and discriminators (word features) are time-dependant. That is to say, the topical meaning of discriminator shifts over time, and a topic has different representations at different times.

We first compute the posterior probability of assigning a new document vector $\mathbf{d}$ to cluster c_j as

$$\begin{aligned} P(c_j|\mathbf{d}, t) &= \sum_{x \in F} P(c_j|x, \mathbf{d}, t)P(x|\mathbf{d}, t) \\ &= \sum_{x \in X} P(c_j|x, \mathbf{d}, t)P(x|\mathbf{d}, t) + \sum_{x \in F \setminus X} P(c_j|x, \mathbf{d}, t)P(x|\mathbf{d}, t) \\ &= \sum_{x \in X} \frac{P(\mathbf{d}|c_j, x, t)}{P(\mathbf{d}|x, t)} \cdot P(c_j|x, t)P(x|\mathbf{d}, t) + \sum_{x \in F \setminus X} P(c_j|\mathbf{d}, t) \cdot \frac{f_{x,d}}{|d|}, \end{aligned}$$

where for non-discriminative features, $P(x|\mathbf{d}, t)$ is estimated as $f_{x,d}/|d|$, since non-discriminative features like stopwords are independent of time. Similar to the offline model, $\sum_{x \in F \setminus X} f_{x,d}/|d|$ is estimated as a constant and denoted by R , and $P(c_j|x, \mathbf{d}, t) = P(c_j|\mathbf{d}, t)$, both only for non-discriminative features. By assuming⁵ $P(\mathbf{d}|c_j, x, t)/P(\mathbf{d}|x, t) = 1$ for discriminative features X , we have

$$P(c_j|\mathbf{d}, t) = \frac{1}{1-R} \sum_{x \in X} P(c_j|x, t)P(x|\mathbf{d}, t). \quad (\text{Eq. 4.13})$$

Similarly, the point assignment of document vector $\mathbf{d}$ is given by $\arg \max_{c_j} P(c_j|\mathbf{d}, t)$, and its topic distribution $P(c|\mathbf{d}, t)$ is learned via one additional normalization step (with the normalization factor R_C) over all topics (including the future new topics) as

$$P(c|\mathbf{d}, t) = \frac{1}{R_C} \sum_{x \in X} P(c|x, t)P(x|\mathbf{d}, t), \quad (\text{Eq. 4.14})$$

so that $\sum_{j=1}^k P(c_j|\mathbf{d}, t) = 1$.

For online topic detection, it is in general not practical to generate the topic distribution over all topics as the point assignment to new (unseen) topics is hard to estimate. Instead, we simply compare and use the raw probability defined in Eq. 4.13.

Estimation of Online Discriminative Model

We assume there are no explicit dependencies between document d and time t given the temporal descriptor x , as shown in the Bayesian network in Figure 4.1. The conditional independence between d and t is reasonable because given d , its publication time t is also given explicitly, and its generative probability only relies on the bag of words, which leads to $P(\mathbf{d}, t|x) = P(\mathbf{d}|x)P(t|x)$.

⁵This assumption is also used in the offline discriminative model.

Figure 4.1: Explicit dependencies among document (d), descriptor (x), and time (t).

In the following, we use all seen documents assigned to c_j as its representation,

$$\begin{aligned}
 P(c_j|x, t) &= \frac{P(c_j, x, t)}{P(x, t)} = \frac{P(x|c_j, t)P(c_j|t)}{P(x|t)} \\
 &= \frac{\frac{1}{N(c_j, \tau)} \sum_{d' \in c_j(\tau)} P(x|\mathbf{d}', t)P(c_j|t)}{P(x|t)}, \tag{Eq. 4.15}
 \end{aligned}$$

where τ indicates the time period $t' \leq t$, $N(c_j, \tau)$ is the number of documents in cluster c_j up to time t .

Inserting Eq. 4.15 into Eq. 4.13, we have

$$P(c_j|\mathbf{d}, t) = \frac{1}{R_C} \cdot \sum_{x \in X} \frac{\frac{1}{N(c_j, \tau)} \sum_{d' \in c_j(\tau)} P(x|\mathbf{d}', t)P(c_j|t)}{P(x|t)} P(x|\mathbf{d}, t). \tag{Eq. 4.16}$$

It is easy to estimate both $P(x|t)$ and $P(c_j|t)$ as follows:

$$\begin{aligned}
 P(x|t) &= \frac{\sum_{d' \in D(t)} TF'(x, d')}{N(t)}, \\
 P(c_j|t) &= \frac{N(c_j, \tau)}{N(\tau)},
 \end{aligned}$$

where $TF'(x, d')$ is the normalized term frequency of word x in document d' , as given in Equation Eq. 4.12, $N(t)$ is the number of documents at time t , $N(\tau)$ is the number of documents up to time t . Apparently, $1/\sqrt{P(x|t)}$ is the IDF and $P(c_j|t)$ could be used to normalized the topic size.

The remaining task is to derive $P(x|\mathbf{d}, t)$ or $P(x|\mathbf{d}', t)$, both of which share the same form. Intuitively, we can treat $P(x|\mathbf{d}, t)$ as a dynamic term frequency that incorporates the temporal information into the static term frequency, $P(x|\mathbf{d})$. Following the Bayes formula, we have

$$\begin{aligned}
 P(x|\mathbf{d}, t) &= \frac{P(x, \mathbf{d}, t)}{P(\mathbf{d}, t)} = \frac{P(\mathbf{d}, t|x)P(x)}{\sum_{x' \in F} P(\mathbf{d}, t|x')P(x')} = \frac{P(\mathbf{d}|x)P(t|x)P(x)}{\sum_{x' \in F} P(\mathbf{d}|x')P(t|x')P(x')} \\
 &= \frac{\frac{P(x|\mathbf{d})P(\mathbf{d})}{P(x)} \frac{P(x|t)P(t)}{P(x)} P(x)}{\sum_{x' \in F} \frac{P(x'|\mathbf{d})P(\mathbf{d})}{P(x')} \frac{P(x'|t)P(t)}{P(x')} P(x')} = \frac{P(x|\mathbf{d})P(x|t) \frac{1}{P(x)}}{\sum_{x' \in F} P(x'|\mathbf{d})P(x'|t) \frac{1}{P(x')}},
 \end{aligned}$$

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

where $P(x|\mathbf{d})$ is the static TF, $P(x|t)$ is the inverse IDF at time t , and $P(x)$ is the normalized cumulative inverse IDF up to time t , defined as follows,

$$P(x) = \frac{\sum_{d'' \in D(\tau)} TF'(x, d'')}{N(\tau)}.$$

We can further simplify $P(x|\mathbf{d}, t)$ to be

$$P(x|\mathbf{d}, t) = \frac{P(x|\mathbf{d})P(x|t)\frac{1}{P(x)}}{\sum_{x' \in X} P(x'|\mathbf{d})P(x'|t)\frac{1}{P(x')} + R_S},$$

since $R_S = \sum_{x' \in F \setminus X} P(x'|\mathbf{d})P(x'|t)\frac{1}{P(x')}$ can be a constant for non-discriminative features.

Similarly, by defining the *dynamic term frequency* and *dynamic inverse document frequency* of x as

$$\begin{aligned} TF''(x, d) &= P(x|\mathbf{d}, t), \\ IDF''(x) &= \frac{1}{\sqrt{P(x|t)}}, \end{aligned} \quad (\text{Eq. 4.17})$$

$P(c_j|\mathbf{d}, t)$ can be reformulated as

$$\begin{aligned} P(c_j|\mathbf{d}, t) &= \frac{1}{R_C} \cdot \sum_{x \in X} \frac{1}{N(\tau)} \left[\sum_{d' \in c_j(\tau)} TF''(x, d') \cdot IDF''(x) \right] \cdot [TF''(x, d) \cdot IDF''(x)], \\ &= \frac{1}{R_C} \cdot \frac{N(c_j, \tau)}{N(\tau)} \cdot \sum_{x \in X} \left[\frac{1}{N(c_j, \tau)} \sum_{d' \in c_j(\tau)} TF''(x, d') \cdot IDF''(x) \right] \cdot [TF''(x, d) \cdot IDF''(x)], \\ &= \frac{1}{R_C} \cdot \frac{N(c_j, \tau)}{N(\tau)} \cdot \mathbf{d}^T \mathbf{c}_j(\tau), \end{aligned} \quad (\text{Eq. 4.18})$$

where $\mathbf{d}$ is the document vector and $\mathbf{c}_j(\tau)$ is the topic mean vector up to time t as follows,

$$\mathbf{c}_j(\tau) = \frac{1}{N(c_j, \tau)} \sum_{d' \in c_j(\tau)} \mathbf{d}', \quad (\text{Eq. 4.19})$$

and each dimension of the document vector is written as

$$w_i = TF''(x_i, d) \cdot IDF''(x_i).$$

From Eq. 4.18, we see that by considering only discriminative features, the online discriminative model is equivalent to a variation of incremental TFIDF clustering, with the following characteristics:

- (i) Both TF and IDF are time-dependant. The TF part considers the generative probability of a given document and the IDF part accounts for the document frequency (DF), both at the current time point;

- (ii) Topics must be global variables until the current time, but they could have *zero* probabilities at the early stages. Once they appear, they will never vanish;
- (iii) Larger clusters have a higher affinity to new documents compared to smaller ones.

Application to Online Topic Detection

Following the same new topic detection strategy as Single-Link-All, for each new document, we first compute the corresponding discriminative probabilities with respect to existing clusters. The raw posterior cluster probability in Eq. 4.13 is used to evaluate a new document’s cluster membership normalized by cluster size, which helps to balance the playing field for cluster of all sizes. The constant R is estimated using a bag of well-known stopwords. If the maximum probability is below a user defined threshold ϵ , a new topic is created and its topic vector is set to be the new document’s vector. Again, global clustering is conducted periodically to ensure a decent clustering quality.

4.4.3 Explore Discriminative Features

Both the offline and online DIS models make one key assumption, that documents can be represented by topic-discriminative features. However, there is no general definition for a discriminative word. We can use Linear Discriminative Analysis [McL04] or other techniques to find discriminators, and compare their performances on topic detection accordingly. However, this is beyond the scope of this dissertation. In Chapter 3, we have effectively used bursty word features as discriminative features for topic detection. Here, we shall again treat bursty words as discriminative words.

The Word Burstiness Phenomenon in News Stream

To illustrate the word *burstiness* behavior for the TDT3 dataset⁶, we consider the following time series plots of document frequency for three word examples as shown in Figure 4.2: a common word *the*, a rare word *acrid*, and a topical word *hurricane* highly related to the topic “Hurricane Mitch”.

From Figure 4.2, we see that only the topical word *hurricane* exhibits burstiness behavior as expected, and its bursts actually correlates with two Hurricane topics: “Hurricane George in Oct, 1998” and “Hurricane Mitch in Nov, 1998”. We thus hypothesize that, there exists

⁶TDT3 is the dataset used in this chapter.

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

Figure 4.2: Document frequency evolutions for three words. The x-axis shows the publication date, and the y-axis gives the normalized document frequency at each day. $N_x(t)$ is the number of documents containing word x at time t , and $N(t)$ is the number of documents at time t .

three categories of words in a text stream: common words, rare words, and bursty words, of which only bursty words are *potentially* discriminative with respect to latent topics. For better demarcation of burstiness behavior, we define a document frequency change score measuring the normalized document frequency change at each time unit compared to the average document frequency as

$$\frac{\left| \frac{N_x(t)}{N(t)} - \frac{\overline{N(x)}}{\overline{N(t)}} \right|}{N_x} \quad (\text{Eq. 4.20})$$

Ideally, the three types of words (common, rare, and bursty) should each have distinct score bands, i.e., the score for common words should not vary much while that of rare words is expected to fluctuate. As a trial, we separate the corpus vocabulary into three categories: 148 words with more than 20% document coverage, 28,463 words with less than 0.5% document coverage, and another 2,574 words with coverage that lies in between the two ends. Figure 4.3 illustrates such a phenomenon, where the y-axis tracks the score averaged over all words in the same category.

From the figure, we see that bursty words actually behave quite differently from common and rare words; the former show moderate normalized document frequency score variation, and are therefore potentially more discriminative. Since bursty words are relatively straight-

Figure 4.3: The burstiness of common, bursty, and rare words in the TDT3 dataset. The x-axis is the date, and the y-axis is the variation of average document frequency score over all words in the same category.

forward to detect and makes good intuitive sense, we shall adopt them as discriminatory word features for detecting latent topics.

Incorporating Bursty Words

Bursty word identification from text streams have recently been investigated by a number of researchers [Kle02, ZS03, FYYL05, HCL07]. Since the goal of this chapter is to utilize bursty words and not to develop a new bursty word identification algorithm, we simply adopt the word trajectory energy approach proposed in Chapter 3 to identify the bursty weight of each word.

We treat each and every word as a document frequency trajectory, i.e., $y_x = [y_x(1), y_x(2), \dots, y_x(T)]$, where each element $y_x(t)$ is a measure of word feature x at time t , which could be defined using the normalized DF score

$$y_x(t) = \frac{N_x(t)}{N(t)},$$

where $N_x(t)$ is the number of documents (local DF) containing word x at day t , and $N(t)$ is the total number of documents for day t .

we decompose the word trajectory $y_x = [y_x(1), y_x(2), \dots, y_x(T)]$ into a sequence of T

complex numbers $[X(1), \dots, X(T)]$ via the discrete Fourier transform (DFT):

$$X(k) = \sum_{t=1}^T y_x(t) e^{-\frac{2\pi i}{T}(k-1)t}, \quad k = 1, 2, \dots, T.$$

We define the word trajectory energy as the bursty weight until time T , by using the dominant power spectrum (DPS) of a given word feature x

$$b_x(T) = \|X(k)\|^2, \quad \text{with } \|X(k)\|^2 \geq \|X(j)\|^2, \quad \forall j \neq k. \quad (\text{Eq. 4.21})$$

Given the word corpus F , where μ_b and σ_b are the mean and standard deviation of bursty weights of all words respectively, the normalized bursty weight $b'_x(T)$ is written as,

$$b'_x(T) = \frac{1 + \frac{2}{\pi} \arctan \frac{b_x(T) - \mu_b}{\sigma_b}}{2}.$$

Table 4.3 lists the raw bursty scores for the three sample words: *hurricane*, *the*, and *acrid*.

Table 4.3: Bursty score examples.

word	$b_x(T)$
hurricane	9.1608
the	0.0990
acrid	0.0011

Comparing their trajectories as shown in Figure 4.2, we see that the bursty word *hurricane* has a significantly larger bursty score, while both common words like *the* and rare words like *acrid* have very small bursty scores.

For offline topic detection, the simplest way of simulating the discriminative features is to set a normalized bursty score threshold. Words with normalized bursty scores above the threshold are chosen as the discriminative features. Such a threshold can be empirically determined. For online topic detection, in the early stage the bursty scores of words are not accurate. We thus use a simple heuristic to enhance discriminative words by incorporating the bursty score into the original cumulative TFIDF score as $TFIDF + \lambda \times b'_x(T)$, where the optimal parameter λ can be estimated via cross-validation on a subset of the seen data.

4.5 Performance Evaluation

4.5.1 Dataset and Data Preprocessing

We use the standard TDT3 dataset, which is one of the few publicly available news datasets with both class labels and timestamps released by the TDT community, as our testbed. The TDT3 dataset includes 51,183 news documents collected during the three month period (92 days) of October through December 1998. Among these, 37,526 English documents originated from 8 English sources. We extracted all on-topic English news documents as **TDT3-Single**, which contains 6,502 documents covering 116 topics, with each document labeled with a single topic. We extracted the other 928 on-topic English news documents as **TDT3-Multiple** covering 73 topics where each document belongs to multiple topics.

We used **TDT3-Single** as the testbed for both offline and online topic detection (hard clustering), and **TDT3-Multiple** as the testbed for offline soft clustering. The distribution of document count in each topic for **TDT3-Single** is shown in Figure 4.4.

Figure 4.4: Topic distribution of documents in **TDT3-Single**.

TDT3-Single is rather unbalanced, wherein only 60 topics contain more than 20 documents, and 15 topics contain more than 100 documents. In **TDT3-Single**, new topics are created continuously from the first day until the 85th day. The cumulative count of topics over time is depicted in Figure 4.5.

Figure 4.5: The cumulative count of topics versus day for **TDT3-Single**.

In **TDT3-Multiple**, the majority of the documents (90%) belong to exactly two topics and the remaining documents are labeled with up to 6 topics. The topic distribution is illustrated in Table 4.4.

Table 4.4: Distribution of documents spanning two or more topics **TDT3-Multiple**.

# topics	2	3	4	5	6
# documents	837	84	6	0	1

After stemming, 36,521 distinct words (set F) are retained in **TDT3-Single** and **TDT3-Multiple** without removing stopwords. A total of 1,473 discriminative words are identified over the whole corpus by setting the normalized bursty score threshold to 0.1; this is the discriminative feature that will be used in our DIS model for offline topic detection. For online topic detection, the parameter λ was estimated to be around 1. Version 2 of the open source indexing software Lucene [luc] was used to tokenize/index the documents and generate the document-word vector.

4.5.2 Experiment 1: Offline Topic Detection - Hard Partition

Methodology

In this section, we discuss the practical method for offline topic detection using topic models. Empirical studies [BDGS03] have shown that for a well-separated and balanced dataset, similar performance can be obtained across different clustering algorithms, with the best

performance corresponding to the true number of clusters. However, in practice, different algorithms typically produce varying clustering results since one or more of the above conditions are easily violated. For example, as shown in Figure 4.4, **TDT3-Single** is an extremely unbalanced dataset (the 15 largest topics accounts for 60% of the documents), and not well-separated (similar topics co-exist, i.e., *Hurricane Mitch* and *Hurricane George*). Therefore, it is not easy to fully recover the topical structure of **TDT3-Single** just by clustering. The question to ask is thus this: *if we cannot recover the full $|Z|$ classes, what is the largest $k < |Z|$ clusters we can find that best resemble the k underlying classes?* Using the intuition that largest topics are more likely to correspond to important events that are of interest to data analysts, our less ambitious goal is to find the largest $k < |Z|$ clusters via incremental topic detection models, i.e., increase k gradually until certain conditions are satisfied.

Furthermore, all topic models are naturally sensitive to the selection of k (number of clusters), which is unknown in topic detection. In fact, the determination of the optimal k remains an open research problem. Many methods have been proposed to decide the optimal k for clustering [JMF99], among them, we adopt a simple yet popular method, which produces the optimal k by identifying a “knee” in the MSE (mean squared error) vs. k plot [JD88]. We use the average internal similarity (which increases monotonically as a function of k) to replace the MSE (which decreases monotonically as a function of k). The average internal similarity is also known as the clustering objective function.

Let $I(k)$ be the average internal similarity for any k -component topic detection model,

$$I(k) = \frac{1}{N} \sum_{j=1}^k \sum_{d \in c_j} \mathbf{d}^T \mathbf{c}_j,$$

our Offline topic detection algorithm for recovering topical structure of a news corpus via an unsupervised approach is shown in Algorithm 4.1. The algorithm utilizes the variations in the unsupervised/internal metric, i.e., internal similarity, to terminate the clustering process. The termination condition is at k where the gradient of $I(k)$ does not change significantly, i.e., the change is less than a user predefined threshold ϵ . This condition attempts to find an optimal top k topics (at k , the cohesiveness of topics approaches a local maxima). The output of the algorithm is a list of clusters, $\{c_1, c_2, \dots, c_k\}$. When k is relatively small, typically the $k - 1$ clusters will correspond to the top $k - 1$ largest topics,

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

with the single remaining large cluster containing a mix of all the other (smaller) topics. The single large cluster is easily found as the disproportionately large cluster with the lowest measure of internal similarity.

Algorithm 4.1 (Offline Topic Detection) (D, ϵ)

Input: News stream D , and threshold ϵ ;

Output: top k largest clusters;

```

1:  $k \leftarrow 0$ ;
2: repeat
3:    $k \leftarrow k + 1$  (increment set manually);
4:   conduct  $k$ -component topic models on  $D$ ;
5:   compute  $I(k)$ ;
6: until  $\Delta I(k) < \epsilon$ .
7: Roll Back the clustering results to  $k \leftarrow k - 1$ ;
8: RETURN  $\{c_1, c_2, \dots, c_k\}$ .
```

We evaluate four topic models: SPK, vMF, LDA, and DIS. The first three models are implemented based on the original C++ source codes of spkmeans, moVMF⁷ and LDA⁸. The parameter κ is initialized to 1 for vMF. For LDA, we reconstruct the topic vector (originally a distribution over words) as a mixture of documents according to the soft-clustering results, so that we can compute the internal similarity properly. DIS was implemented in Java. By increasing k in increments of 10 from 10 to 116, we conducted 11 rounds of hard partitioning on **TDT3-Single**. All models except LDA utilized the same method for initializing the cluster centroids. Results were averaged over 10 runs.

Evaluation Metrics

We adopted supervised/external measures to objectively evaluate the quality of topics generated by the different topic detection models since the true document topic labels are known (but not used in the detection process). There are various external measures such as confusion matrix, classification accuracy, F1 measure, average purity, average entropy, and mutual information, etc., as described in [Gho03]. We selected five of the measures for offline topic detection in **TDT3-Single** (hard partition): *Cluster Purity*, *Cluster Entropy*, *Class Purity*, *Class Entropy*, and *normalized Mutual Information (nMI)*. The first four purity/entropy measures collectively cover clustering performances. Different topic models

⁷Available from <http://www.cs.utexas.edu/~suvrit/work/progs/movmf/>, implements both spkmeans and moVMF.

⁸Available from <http://www.cs.princeton.edu/~blei/lda-c/>, which implements the variational inference of all parameters required by LDA.

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

typically have different biases for each of the measures. The normalized mutual information nMI was proposed as a superior alternative over purity/entropy by [SG02]. It accounts for both entropy and purity by calculating the mutual information between two random variables: cluster label c and class label z . It follows that the nMI score of c and z will be high if they are generated by the same distribution. In general, the larger the purity/nMI value the better the topic, and a good model should have a low entropy.

Assume that $|C|$ clusters (topics) are generated for dataset D , in which there are $|Z|$ true classes, a contingency table $Y(z, c)$ can be produced, as shown in Table 4.5.

Table 4.5: A sample contingency table (2×3) for topic detection.

<i>class \ cluster</i>	c_1	c_2	c_3	sum
z_1	$Y(z_1, c_1)$	$Y(z_1, c_2)$	$Y(z_1, c_3)$	$N(z_1)$
z_2	$Y(z_2, c_1)$	$Y(z_2, c_2)$	$Y(z_2, c_3)$	$N(z_2)$
sum	$N(c_1)$	$N(c_2)$	$N(c_3)$	N

Based on Table 4.5, we define the following measures. Cluster Purity $P(C)$ is defined as the weighted sum (average) of individual cluster purities

$$P(C) = \frac{1}{N} \sum_{j=1}^{|C|} \max_i Y(z_i, c_j).$$

Cluster Entropy $H(Z|C)$ measures the diversity of clusters, and is defined as

$$H(Z|C) = - \sum_{i=1}^{|Z|} \sum_{j=1}^{|C|} \frac{Y(z_i, c_j)}{N} \log \frac{Y(z_i, c_j)}{N(c_j)}.$$

Class Purity $P(Z)$ computes the average precision of topics as

$$P(Z) = \frac{1}{N} \sum_{i=1}^{|Z|} \max_j Y(z_i, c_j).$$

Class Entropy $H(C|Z)$ measures the average recall of topics as

$$H(C|Z) = - \sum_{i=1}^{|Z|} \sum_{j=1}^{|C|} \frac{Y(z_i, c_j)}{N} \log \frac{Y(z_i, c_j)}{N(z_i)}.$$

The definition of normalized Mutual Information (nMI) is given below

$$nMI = \frac{I(Z; C)}{\sqrt{H(Z) \times H(C)}} = \frac{H(Z) - H(Z|C)}{\sqrt{H(Z) \times H(C)}},$$

where $H(Z) = - \sum_{i=1}^{|Z|} \frac{N(z_i)}{N} \log \frac{N(z_i)}{N}$ and $H(C)$ is defined similarly.

Result Analysis

In this section, we analyze the hard partition performances of the offline topic detection models on **TDT3-Single**. We first report the time efficiency of each model, followed by in-depth analysis of Algorithm 4.1. Lastly, we evaluate the quality of topics generated by each model.

Figure 4.6 compares the running time of the 4 approaches. We can see that as the number

Figure 4.6: Offline topic detection time complexity versus number of clusters k (log-scaled).

of clusters k increases, the running time of LDA grows exponentially. When k reaches 116, the total running time of LDA is more than 40 hours. The running time of other models grows supra-linearly with k . DIS has the same running time complexity as vMF, both of which are around 10 times faster than SPK. DIS, which uses a smaller number of word features, is much faster than SPK although it uses the same iterative process as SPK. On the other hand, vMF tends to converge very quickly, typically within a few iterations, which makes it very efficient in practice.

After applying Algorithm 4.1, the average internal similarities over different k and their gradient plots are shown in Figures 4.7 and 4.8, respectively.

From the comparisons, we draw the following insights:

- The $I(k)$ curve of LDA is rather smooth, without significant variation in gradient. In other words, LDA is not sensitive to the choice of k . LDA has the highest absolute

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

Figure 4.7: Internal Similarities across various k on offline topic detection. Figure 4.8: Gradient of Internal Similarity across various k on offline topic detection.

internal similarity scores, because many documents are heavily overlapped in terms of topic distribution. For example, many words (not only the common words) like *Clinton* span different topics. This leads to the formation of some large clusters in the early stages (when k is small). For example, LDA will stop at $k = 20$ only if we set a very small similarity threshold ϵ , i.e., 0.01. In general, the offline clustering algorithm cannot be applied effectively for LDA because it will miss the majority of the smaller

topics. This is partially due to the lack of a straightforward way to compute the internal similarity for LDA since the topics are built over words and not directly over document. For a fairer comparison, we subsequently plotted the perplexity gradient trend of LDA, as shown in Figure 4.9. The perplexity is defined as

$$perplexity = \exp\left\{-\frac{\sum_{i=1}^N \log P(\mathbf{d}_i)}{\sum_{i=1}^N |d_i|}\right\},$$

where $P(\mathbf{d}_i)$ is the likelihood of document d_i and $|d_i|$ is the length of d_i . A smaller perplexity value indicates a higher likelihood of the data. Comparing Figure 4.9 with

Figure 4.9: Perplexity gradient across various k for LDA offline topic detection.

Figure 4.8, LDA performs much better in terms of perplexity. There is a knee around $k = 30$ in the perplexity gradient trend, although after that the gradient changes quite randomly. This result also indicates that LDA is more suitable for modeling documents. However, LDA does not produce good clusters, as shown later by the poor quality of clusters generated by LDA, no matter at $k = 20$ or at $k = 30$.

- Both SPK and vMF have similar $I(k)$ curves and corresponding gradient trends. Their performances are *very* sensitive to the choice of k . For example, both will stop at $k = 40$ with a threshold of $\epsilon = 0.05$, and later at $k = 60$ with $\epsilon = 0.025$. The advantage is that clusters are better separated, but selection of an optimal threshold ϵ is difficult.

- The characteristic of DIS lies somewhere between LDA and SPK (vMF); it exhibits clear trend ($I(k)$ and its gradient) variations in the early stages, but cannot discover more topics after a certain point. Its gradient curve clearly indicates that $k = 30$ is a good stopping point; afterwards, increasing k will not improve the clustering results. This verifies our conjecture that bursty scores of words could enhance the cohesiveness of large topics [HCLZ07], at the price of missing out on the smaller topics. The enhancement on cohesiveness of large topics will in turn force large topics to be discovered in the early stages of clustering, i.e., with a small k . Consequently, DIS may be more suitable for detecting large topics, and therefore more useful in practice for finding the top k topics.

Next, we shall evaluate the quality of topics generated by each model, using external measures. Figures 4.10-4.13 list the average purity/entropy comparisons for all models, where each model has its specific bias towards certain measure; and in Figure 4.14, the nMI, which summarizes purity and entropy, is compared.

Figure 4.10: Cluster purity across various k on offline topic detection.

From the results, we can make the following observations.

- Even after reducing the feature space by 96 percent, DIS still slightly outperformed the other models on average, as shown in the nMI comparison in Figure 4.14. Although LDA also performs feature reduction implicitly, it does so in a supervised

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

Figure 4.11: Cluster entropy across various k on offline topic detection. Figure 4.12: Class purity across various k on offline topic detection.

manner requiring predefined set of topics. LDA first generate the topic distributions from the training data. Only those features with high generative probabilities (given topics) will be selected, which is subsequently used for testing. The limitation of LDA is clear: the test data must have the same topic distribution as the training data. To be fair, LDA was never designed for document-level clustering, but was instead proposed for generating unseen documents originating from the same topic distribu-

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

Figure 4.13: Class entropy across various k on offline topic detection. Figure 4.14: nMI across various k on offline topic detection.

tion as the training data, i.e., LDA models topics with short descriptions (words). As we mentioned before, in LDA, documents are heavily overlapped in terms of topic distributions. This is also the main reason why LDA has the worst performance on all measures. On the other hand, DIS performs feature reduction (by selecting bursty features) in a completely unsupervised way without any topical assumption on the data. It is thus more suitable for offline topic detection. SPK performs very similarly

to DIS, and better than vMF at or before $k = 50$. This further verifies that for large topics, a discriminative model is better than more complicated generative models like vMF and LDA, as long as we can find a good set of discriminative features. After $k = 50$, vMF is slightly better than SPK, which means that vMF has a certain ability to generate small topics whose data are incomplete.

- DIS has slightly better cluster purity/entropy (less than 5% improvement on average, as shown in Figures 4.10-4.11), compared to SPK and vMF. On the surface, all three models may appear to produce qualitatively similar clusters. However, if we scrutinize the cluster quality in more details, they actually performed differently. For example, when k is small, both DIS and SPK improve a lot on class purity/entropy (10%-20% improvement on average, as shown in Figures 4.12-4.13), compared to vMF. With the help from bursty words, DIS ensures that each of the large classes stay together as one cluster. To a lesser extent, SPK also attempts to keep large classes as one cluster, while on the other hand vMF tends to split a large class into several smaller clusters. Moreover, as k increases, SPK will also start splitting the larger classes. Since a vMF distribution could represent a smaller topic better than just using a few sample points, vMF leads the pack in maintaining the cohesiveness of small topics. This result actually provides valuable insights on how offline topic detection models can be personalized: if users are more interested in mainstream topics, using a discriminative model like DIS will be more effective; while a generative model like vMF can model smaller and less popular (outlier) topics better.

4.5.3 Experiment 2: Offline Topic Detection - Soft Partition

Methodology

In many cases a document could belong to multiple topics. Given a document d , we would like to generate a set of probabilities $P(c_j|\mathbf{d})$ for various c_j . In general, only the top m probabilities are meaningful. We shall compare the soft partitioning performances between vMF⁹, LDA and DIS. Similar to hard partitioning on offline topic detection, we have to determine the optimal number of clusters k for all three models. In addition, an optimal m has to be determined as well. We could use the same gradient method for both. Since

⁹vMF can be viewed as a generalized version of SPK for soft-clustering.

we have already tested the gradient method in Section 4.5.2, for simplicity, here we only examine the external performances of soft partition by varying m and fixing $k = 73$ to be the correct number of clusters for **TDT3-Multiple**.

Evaluation Metrics

Since a document can be assigned to all clusters now, the contingency table is thus meaningless. Even if we only select top m ($m > 1$) clusters for each document, the classical purity/entropy measures do not capture the ordering information. For example, a low class entropy does not necessarily indicate a high recall, since documents originating from this class may belong to other classes as well. We thus need to devise new external evaluation measures.

Given a document d , although we cannot find one-to-one relationships between its cluster and category labels, we can still compute how many other documents share the same category label with d , which gives us some idea about the overall clustering quality. Let $g(i, j)$ be the number of ground truth categories assigned to both document d_i and d_j , and $h(i, j)$ as the number of clusters containing both documents. Both $g(i, j)$ and $h(i, j)$ are fixed if only the top m clusters are retrieved. Ideally, $h(i, j)$ matches $g(i, j)$ for all pairs of documents. If $h(i, j) = g(i, j)$, we say that the current soft clustering solution correctly clusters d_i and d_j . If $h(i, j) > g(i, j)$, there exists $h(i, j) - g(i, j)$ false alarms. If $h(i, j) < g(i, j)$, $g(i, j) - h(i, j)$ co-existing classes are missed. Accordingly, we can build an overall contingency table, as shown in Table 4.6, where we have

Table 4.6: Contingency table for document pairs.

	co-exist in classes	do not co-exist in classes
in the same cluster	a	b
not in the same cluster	c	d

$$\begin{aligned}
 a &= \sum_{i=1}^N \sum_{j=i+1}^N \min(h(i, j), g(i, j)), \\
 b &= \sum_{i=1}^N \sum_{j=i+1}^N \min(h(i, j) - g(i, j), 0), \\
 c &= \sum_{i=1}^N \sum_{j=i+1}^N \min(g(i, j) - h(i, j), 0).
 \end{aligned}$$

Finally, based on Table 4.6, we can define the precision (P), recall (R) and F-measure (F) in the following,

$$P = \frac{a}{a + b},$$

$$R = \frac{a}{a + c},$$

$$F = \frac{2PR}{P + R}.$$

Given Table 4.6, we can completely replicate the soft clustering results (proof omitted). Therefore, the above external measurement is a reasonable estimate of clustering quality.

Result Analysis

The comparison on time efficiency is quite similar to hard partitioning and is omitted here. Figure 4.15 illustrates the precision/recall by varying m from 1 (top/left) to 10 (bottom/right). Further increasing m will reduce the precision while recall remains at 1. Not surprisingly, all models achieved the best precision/recall compromise at $m = 2$ since

Figure 4.15: Precision and recall across various top m soft clusters.

about 90% of the documents belong to 2 topics as shown in Table 4.4. Afterwards, the precision drops significantly with the recall saturating at 1.

Furthermore, from Figure 4.15, we see that both DIS and vMF performed much better than LDA. For example, when $m = 2$, the weighted harmonic mean of precision and recall, or the F-measure, of DIS (vMF) is 0.7970 (0.7372), which is more than twice that of LDA (0.3061), as shown in Figure 4.16.

Figure 4.16: F-measure across various top m soft clusters.

It is not hard to understand why DIS and vMF showed vast improvements over LDA on soft clustering; we evaluated the soft clustering performance based on the multiple categorical labels of documents, not words. On average, DIS outperforms vMF but the improvements are always less than 5% across m . This result is a bit surprising. We originally thought that bursty scores of words should significantly enhance soft clustering since multiple topic labels assigned to a document are usually related and would be collectively enhanced by common bursty words. It turns out in practice that bursty words actually negatively impact soft clustering, which can be illustrated in the following example. Suppose two topics are well-separated based on the standard TFIDF weighting. But after incorporating bursty scores of words, the two topics may be pulled closer together if they share many common bursty words. The end result could boost the similarity of a previously unrelated topic that happens to have a few common bursty words with a document, to the extent of surpassing the legitimate topics. Fortunately, this side-effect is not too prevalent and only served to decrease the overall improvements slightly.

In conclusion, bursty scores of words contribute to maintaining large topics in hard clustering (good class purity/entropy), but also yields the overall positive effect of pulling together similar topics in soft clustering.

4.5.4 Experiment 3: Online Topic Detection

Methodology

Compared to offline topic detection, online topic detection is a more challenging and interesting problem. Our goal here is to devise an efficient and effective algorithm for on-line topic detection. We shall adopt the non-probabilistic model, Single-Link-All, as our topic-detection strategy [All02b]. The major problem of deploying Single-Link-All in practice [AHF⁺05] is that the size of detected cluster is typically either too large or too small. In other words, cluster quality is usually very poor. Allan *et al.* [AHF⁺05] suggested two possible remedies: *hierarchical clustering* and *increasing the threshold ϵ* . Both solutions could break large clusters into pieces. However, to avoid producing too many small clusters, additional strategies should be considered at the same time.

We focus on the second remedy since we deal only with a flat topic structure. Increasing the threshold ϵ will increase the difficulty of inserting a new document into an existing cluster. Accordingly, it is easier to create new clusters, as well as false alarms. Our target is then very clear: minimize the number of false alarms. Allan *et al.* [AHF⁺05] further considered two engineering solutions for reducing the false alarms. First, the clusters have “age” so that it is difficult to add new documents to an “aged” cluster. Second, each existing cluster is represented by its averaged centroid (not corresponding to any real document) rather than the single most representative document. Accordingly, we define two more baseline models, *Single-Link* and *Incre-Mean*, for online topic detection. In Single-Link, every new document is only compared to documents published within the last 7 days. In Incre-mean, each cluster is represented by its centroid, which is continually updated with new member documents. Clusters that have not been updated for a certain time period, i.e., 7 days, will be effectively discarded and archived, and not considered further during new topic detection. Every new document is thus compared to the centroid of a reasonably fresh existing cluster.

All three baseline models incrementally increase the number of clusters k , but none of them perform any global optimizations at any time. For practical online topic detection,

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

some modifications are necessary. First, global clustering is periodically conducted on the latest set of documents. Second, the number of clusters k has to be reduced to remove false alarms. These modifications are summarized in Algorithm 4.2, which will be used for the all online topic detection models. Algorithm 4.2 is indeed an extension of Single-Link and

Algorithm 4.2 (Online Topic Detection) (D, ϵ)

Input: News stream D , and threshold ϵ ;

Output: detected (new) clusters on the fly;

```

1:  $k = 1, c_k \leftarrow d_1, C \leftarrow c_k$ ;
2: announce a new topic  $c_k$ ;
3: repeat
4:   if a new day begins then
5:     for  $i = 1$  to  $k$  do
6:       if  $c_i$  has not been updated for the past 7 days then
7:          $C = C - c_i, k = k - 1$ ;
8:       end if
9:     end for
10:    run  $k$ -component topic detection model on the past 7-day data,  $I$  records the
        average internal similarity of the clustering results.
11:     $max = k, min = 0$ ;
12:    repeat
13:       $k = (max + min)/2$ ;
14:      run  $k$ -component topic detection model on the past 7-day data,  $I'$  records the
        average internal similarity.
15:      if  $I' > I$  then
16:         $max = k$ ;
17:      else
18:         $min = k$ ;
19:      end if
20:    until  $k$  does not change;
21:  end if
22:  Compute the similarity of new document  $d$  with the  $k$  existing cluster centroids (or,
    compute the posterior probabilities of  $k$  existing topics given  $d$ );
23:  if the largest similarity is greater than  $\epsilon$  then
24:    assign  $d$  to the nearest cluster; update this cluster's parameters;
25:  else
26:     $k = k + 1, c_k \leftarrow d$ ;
27:    announce a new topic  $c_k$ ;
28:  end if
29: until there are no more new documents.

```

Incre-mean, with two major changes: 1) a global clustering is conducted every day; 2) a simple “divide and conquer” strategy is adopted to reduce the number of clusters k .

Evaluation Metrics

The performance of online topic detection largely relies on the threshold ϵ . We adopted the Detection Error Tradeoff (DET) curve [MDK⁺97] which has been widely used in TDT [KA04] to measure the miss and false alarm values at each threshold. Tracing the DET curve, TDT defines the official evaluation measure as a cost function, which is a weighted combination of miss and false alarm values, as follows [AHF⁺05],

$$Cost = C_{miss}P(miss)P(target) + C_{fa}P(fa)P(off - target), \quad (\text{Eq. 4.22})$$

where $P(target) = 1 - P(off - target)$ is the prior probability that a document will be a new topic (0.02, derived from the TDT training data), $C_{miss} = 10$ and $C_{fa} = 1$ are user-specified penalty factors, and $P(miss)$ and $P(fa)$ are the actual miss and false alarm values. The cost function is further normalized because the system would get a default score of 0.2 if it fails to detect any new topic, and a score of 0.98 if it accepts each and every new document as the new topic. The final cost is divided by 0.2, indicating the system with a detection cost of 1.0 is no better than a system that does not detect any new topic.

The above TDT evaluation metric actually still allows for a large proportion of misses, despite it already penalizing misses more than false alarms. We have discussed before that increasing the threshold ϵ could help remove those overly large clusters. In fact, increasing the threshold ϵ could also reduce the miss rate, at the cost of producing more false alarms. In practice, we are more interested in whether topic models could reduce the false alarms under a very low miss rate, i.e., $P(miss) = 0$.

Result Analysis

We only use **TDT3-Single** as the testbed for online topic detection as it is neither straightforward nor meaningful to evaluate the miss/false alarm for a document with multiple topic labels. Figure 4.17 shows the running time of the various online topic detection models.

As the threshold ϵ increases, the non-probabilistic models tend to be constant because: 1) all documents are detected as new topics (false alarm rate is maximized) after a certain value of ϵ , i.e., for Single-Link-All; 2) the number of documents used for comparison maintains stable after introducing the sliding window (size of 7 days), i.e., for Single-Link and Incre-mean. Single-Link and Incre-mean are the fastest methods among all models because no iterations are involved and the fewest number of documents (within the sliding window)

Figure 4.17: Online topic detection time complexity (log-scaled).

need to be compared. On the contrary, the running time of probabilistic models grows significantly as the number of clusters k increases (induced by the increment of ϵ), except for vMF which converges so quickly that it is comparable to non-probabilistic models. More specifically, the online running time of LDA increases the most. Without feature selection for online topic detection, DIS also needs considerable time. Its discrete probabilistic topic assignment required a bit more time compared to the point topic assignment adopted by SPK. However, considering that a reasonably small value of ϵ is often selected in practice, both DIS and SPK are even faster than the traditional Single-Link-All method. Therefore, we conclude that our proposed DIS probabilistic model combined with Algorithm 4.2 to be highly effective for online topic detection.

Figure 4.18 illustrates the log-scaled DET curve for all models on **TDT3-Single**. The DET curves in Figure 4.18 are a bit complicated. In the upper left region, Single-Link-All and Incre-mean have the best miss/false alarm balance. Next, DIS and Single-Link lead the pack respectively. Finally, DIS has the lowest false alarms under very low miss rates. That is to say, different models have their own bias/edges at different DET regions. For example, Incre-mean works well under a small threshold ϵ (very few false alarms) by producing the largest number of correct topics, and DIS has the best performance under a large threshold ϵ (very low miss rate), where the fewest number of false alarms are produced. The other three models, SPK, vMF and LDA, always perform poorly everywhere along the DET curve.

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC DETECTION

Figure 4.18: The DET curve for online topic detection.

Table 4.7 lists the minimal DET cost values of all seven models. Not surprisingly, DIS

Table 4.7: Minimal DET cost values of all models for online topic detection.

	Single-Link-All	Single-Link	Incre-mean	
cost	0.4326	0.4133	0.4702	
	SPK	vMF	LDA	DIS
cost	0.7680	0.7069	0.8428	0.4008

achieved the minimal¹⁰ cost value of 0.4008, followed by Single-Link which achieved a close second at 0.4133. Comparing Single-Link with Single-Link-All, we see that the sliding window did not increase the cost (even a slight improvement), yet contributed a lot in speeding up the probabilistic models (it largely reduced the number of documents needed for clustering). Incre-mean did not improve the performance, which is consistent with earlier finding that representing clusters of documents by their centroid is ineffective [All02b]. This further verifies our hypothesis that a global clustering process is necessary from time to time. Although DIS has not improved a lot on the cost value, it achieved a much smaller false alarm rate (less than half of the others) under zero miss rate, as shown in Table 4.8.

Table 4.8: False alarms under *zero* miss rate of all models for offline topic detection.

	Single-Link-All	Single-Link	Incre-mean	
false alarm (%)	63.97	49.83	62.72	
	SPK	vMF	LDA	DIS
false alarm (%)	85.08	61.42	94.31	24.15

All in all, Algorithm 4.2 can significantly reduce false alarms by merging small clusters, and periodic global clustering based on flat topic models can enhance the overall clustering for online topic detection. However, partially due to the simplicity of Algorithm 4.2, not all topic detection models could achieve the goal. For example, if we only consider cumulative TFIDF without paying attention to the DF or the bursty properties of words/topics, SPK and vMF both fail to cluster documents well on the fly, which will often lead to huge clusters. The LDA topic model, on the other hand, performed poorly for online topic detection since it was primarily designed for offline word clustering.

¹⁰The best TDT system achieves a cost value of 0.3 with 28% miss rate and a 0.3% false alarm rate on average [AHF⁺05].

4.6 Discussions

No model works best across all types of data and applications. It is therefore important to select the most suitable model for the problem at hand. Non-probabilistic models are based on the TFIDF heuristic [SB88], which weighs the importance of each word in a document (proportional to the number of times the word appears in the document but offset by the document frequency of the word in the corpus). Weighted by TFIDF, documents originating from the same topic are similar to each other and will be clustered together, *assuming* these documents have high weights on a common set of words. Variations of TFIDF are often developed for different applications, but the vast majority of them are lacking in theoretical justification. Through a careful analysis on the posterior probability of topic variables for the discriminative model DIS, we formally derived the temporal TFIDF, which is effective for both offline and online topic detection. With the temporal TFIDF model, the posterior probability can be optimized via the standard clustering iterative process. However, the discriminative models *assume* only discriminative words can be used. Except for the TFIDF heuristic, the vMF mixture models also *assume* that similar documents are sampled similarly from a set of distributions (topics). A proper distribution (i.e, von-Mises Fisher) has to be carefully selected for the data. Moreover, the distributions (topics) often highly overlap with each other. Under this circumstance, if we cannot properly cluster the data, it is hard to expect the mixture model to produce a good result. The vMF mixture model is an effective tool on soft clustering. In addition to the mixture assumption, the LDA hierarchical models further *assume* that topics are sampled independently from the corpus of words. Different combinations of words (including the number of appearance) form different topics. The new assumption brings forward many interesting applications including modeling unseen documents, feature reduction in classification, etc. However, for document-level topic detection, it does not provide additional information on the inter and intra document statistical structure (words are still exchangeable within documents; and similar documents are still tied by common topical words).

The above analyses illustrate that a sophisticated and mathematically elegant topic model may not necessarily work well for topic detection. In practice, the more parameters there are to estimate, the harder it is to obtain accurate probabilistic estimates given the limited amount of training data at hand. For example, Table 4.9 lists the number of parameters required to be estimated by each model, where k is the number of topics expected to

Table 4.9: Model complexity.

models	number of parameters
SPK/Single-Link-All	k
DIS	k
vMF	$2k$
LDA	$k + n \cdot k$

be detected, and n is the size of word feature corpus. LDA contains too many parameters, which are hard to estimate accurately in practice. The vMF model works quite well since it only increases the model complexity slightly. Both the non-probabilistic and our discriminative DIS models share the simplest assumption that discriminative words are required. Thus, they are equally simple and require the fewest number of parameters. Moreover, DIS further provides an elegant mathematical justification to the popular TFIDF model. Since there is no quantified definition on discriminative features, we propose a simple heuristic that uses the burstiness of words to evaluate how discriminative a word is to some topics. Empirical studies on the standard TDT3 dataset indicate that simple models always outperform more sophisticated models for both offline and online topic detection.

4.7 Summary

In this chapter we studied a set of topic detection models on both offline and online document-level topic detection problems for news stream. We first investigated the traditional non-probabilistic models, along with their limitations on topic detection, i.e., no theoretical explanation, documents cannot belong to multiple topics, and it is hard to tune the parameters for online topic detection, etc. We then proposed a discriminative model (DIS) for topic detection in news stream, which is a simple and effective probabilistic model without the assumptions made by more complicated generative models like vMF mixture and LDA. We show the equivalence of DIS to the clustering process of a variation of TFIDF under the condition that only discriminative words are used. A simple heuristic of utilizing the bursty phenomenon of words is used to extract discriminative features. DIS in fact provides a theoretical explanation to the classical non-probabilistic models for topic detection. Moreover, we also benchmarked DIS soft-clustering performance on offline topic detection. The experimental results show that DIS is surprisingly good in recovering large topics from

CHAPTER 4. TEMPORAL DISCRIMINATIVE PROBABILISTIC MODELS FOR DOCUMENT-LEVEL TOPIC
DETECTION

historical news stream (offline topic detection), assigning multiple topics to a document (soft offline topic detection), and reducing the overall false alarm rate (online topic detection). Our results thus lead to the conclusion of this chapter: sophisticated models like vMF or LDA may shine when there are enough training data for accurate parameter estimation, but for the problem of topic detection, a simple and mathematically elegant model like DIS can be surprisingly effective and practical (fast).

Chapter 5

Intra-topic Transition Detection via Classification

In Chapters 3 and 4, we examined discriminative features of news stream and applied them to the detection of topics at the word and document level, respectively. However, within a topic (genre), the intra-structures are still unclear and require further analysis. Since unsupervised methods like clustering cannot reliably separate a topic cluster into a meaningful hierarchy of related topics, we will use a supervised method to learn the discriminative features (support vectors) for sequential segments (events) within a topic. Specifically, we define a new problem called Anticipatory Event Detection (AED) [HCL06a, HCL06b], and propose ways to address it.

5.1 Motivations and Problem Definition

In practice, topics belonging to the same genre (e.g., election of US President, etc.) often involve a common set of event¹ transitions (e.g., nomination of party's Presidential candidates, nomination of party's Vice-Presidential candidates, election of party's Presidential team, election of Presidential team, etc.). Further analyzing such sequential event transition structure within a topic has some obvious real-life importance. First, the definition of topic itself is sometimes unclear, partially because it is hard to define a clear granularity for topics. Understanding the intra-structure of a topic would satisfy different users' interest. Moreover, the iterative intra-topic exploration could potentially be used to build a hierarchy of topics. Second, if the user is only interested in a specific event embedded in a more general topic, it is impractical to bombard users with the all documents from that

¹We use events to refer to specific sub-topics occurring within a topic.

topic. Instead, we would like to alert the user only for the event that he is interested in. For example, we wish to discover and monitor transition of a user predefined news event such as the capture of Osama bin Laden, the result of the US Election, the updated death toll from the 2008 Wenchuan earthquake, the outcome of a single NBA basketball match, etc.

We propose to model the intra-structure of the same topic type as an event transition graph.

Definition 5.1 (Event Transition Graph (ETG)) *An Event Transition Graph (ETG) models multiple event transitions belonging to the same topic genre as a sequence of k events $E = e_1, e_2, \dots, e_k$ related by a set of transition links between each pair of transitive events of the form*

$$t_{i,j}, \text{ if } \exists \text{ transition from } e_i \text{ to } e_j, \text{ where } i < j \text{ and } e_i, e_j \in E.$$

A user can select one of the several event transitions as the anticipatory event transition (AET) or user preference, and would like the first news document reported after the transition to be detected. We call this problem Anticipatory Event Detection (AED), which is defined as follows.

Definition 5.2 (AED) *The objective of AED is to detect and identify entities (messages or documents) that confirm the occurrence of a user specified Anticipatory Event Transition (AET), which is also known as the user preference.*

The idea is to allow a user to receive only predefined news alerts, i.e. those alerts involving a major shift/change in information content, thereby saving users from having to deal with unnecessary interruptions.

5.2 The AED Framework

5.2.1 AED on Document Streams

In practice, AED is usually applied to an online stream of news documents. Figure 5.1 shows a global time-ordered sequence of documents, some on-topic and others off-topic with respect to an AET. Among the on-topic documents, only those that confirm the AET are considered hit documents, and should be identified by an AED system.

Figure 5.1: AED for document streams.

Ideally, a user should be allowed to specify any desired AET explicitly. However, this is not possible in practice due to the lack of a machine-understandable syntax to describe event semantics. One compromise is to present various known and trained ETGs to the user, from which he or she could pick a desired AET and specify the associated named entities. For example, if an ETG on the disposal of country leaders is available, a user could select from it an AET denoting resignation. The user should also supply a set of keywords such as “Taiwan President Chen Shui Bian” to indicate that he wishes to detect events confirming this particular resignation and not every resignation in the world.

Figure 5.2 shows an ETG describing a common structure shared by all company acquisition topics. Suppose a user is interested in the event transition $t_{2,3}$ from event e_2 (“In talks to acquire”) to event e_3 (“Announces acquisition”). As multiple news articles could be associated with the “Announces acquisition” event, the earliest one will be the first story confirming the transition and is therefore the candidate document to be identified by an AED system.

Figure 5.2: An Event Transition Graph (ETG) for the “acquisition” topic genre.

5.2.2 Types of Anticipatory Event Detection

Although the objective of AED is to detect the first story after the user specified AET, it may not always be successful in practice. Here, we formally define four types of AED scenarios.

Suppose we are given a set of N news documents $D = \{d_1, \dots, d_N\}$ about a topic, and a sequence of k events $E = [e_1, \dots, e_k]$ and its associated ETG. Each news article d has a publication date/time represented by $t(d)$ and an event type in E represented by $e(d)$, the latter of which is also known as the true event of d . We assume that all news documents in D are sorted in time ascending order, i.e., $t(d_i) \leq t(d_j) \forall i < j$, and all events in E are sorted in time ascending order, i.e., $t(e_i) \leq t(e_j) \forall i < j$.

By applying any anticipatory event detection technique on a news document d , we obtained its assigned event denoted by $e'(d)$. Given an anticipatory event transition $t_{u-1,u}$ as the user preference, the objective of AED is therefore to find the news document d_m that satisfies:

$$d_m = \arg \min_{d_i} \{t(d_i) \mid \forall d_i \text{ where } e'(d_i) = e_u\}. \quad (\text{Eq. 5.1})$$

To make the time comparison easier between the detected first story d_m and the event e_u , we also define the *true time* of e_u , $t(e_u)$, as follows:

$$t(e_u) = \min \{t(d_i) \mid \forall d_i \text{ where } e(d_i) = e_u\}. \quad (\text{Eq. 5.2})$$

Once the first story d_m of the anticipatory event e_u is determined by the AED classifier, all subsequent news articles, d_j , $j = (m + 1), \dots, N$ will be assigned to event(s) post $t_{u-1,u}$. Since the first story identified by AED may be premature, delayed, or undefined (never found), we define four types of AED scenarios as follows:

- Accurate Alarm* : $t(d_m) = t(e_u)$. First story of e_u found successfully.
- Delayed Alarm* : $t(d_m) > t(e_u)$. First story found was too late.
- False Alarm* : $t(d_m) < t(e_u)$. First story found was premature.
- Miss* : $t(d_m) = \text{undefined}$. No d_i in D has $e'(d_i) = e_u$. AED fails to even identify the event.

Figure 5.3 graphically depicts each of the four types of AED scenarios. In practice, the preferred scenarios are ranked in descending order of preference as follows: accurate alarm, delayed alarm, false alarm and miss. Intuitively, a delayed alarm is preferred over a false alarm or miss according to the age-old saying, “better late than never”.

5.2.3 AED Prototype

Our AED prototype is shown in Figure 5.4. Here, we make a number of simplifying assumptions.

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

Figure 5.3: The four AED scenarios.

Figure 5.4: Online AED system showing only the selected AET of the ETG.

- A reliable ETG is available, from which the user selects a desired AET and enters a set of related key words (usually named entities).
- The system retrieves a set of training documents based on the selected AET. The articles could be based on historically similar topics.
- The training documents are manually annotated either at the sentence or document level as belonging to one of the two possible states of the AET, pre (-) or post (+).

Based on the above assumptions, the system trains a classifier for each AET in the ETG using the labeled/annotated documents. As long as two sets of training documents for both states of a transition are available, a classifier can be pre-trained in offline mode.

After training, the AED prototype operates online as follows:

- (i) The user inputs a set of keywords describing the desired event and selects the appropriate AET from an available ETG.
- (ii) The system monitors an online news stream and filters off a set of candidate documents matching the user specified keywords.
- (iii) The trained classifier corresponding to the user selected AET is applied to this candidate set, where each document is classified as negative or positive. Once a positive document is found, the AED system is deemed to have detected the anticipatory event.

5.3 Modeling the AET

We have experimented with two different resolutions for training the transition model of Figure 5.4, 1) sentence resolution, and 2) document resolution. Training at the sentence resolution is not only hard, but also extremely labor intensive because every sentence has to be manually annotated. In this section, we describe both approaches.

5.3.1 Sentence Classifier

A sentence classification model was initially built to model the AET [HCL06a]. We considered the sentence resolution based on the intuition that the most representative sentences from on-topic AET confirming articles typically provide a good summary of the transpired event transition.

In general, an event transition can be confirmed from a sentence, given enough context. For example, the following sentence would qualify as a “hit” sentence for the anticipatory event “win basketball match”.

- (i) Hit Sentence: *“The Knicks outscored Philadelphia 32-22 in the fourth quarter to secure the win.”*
- (ii) User Preference (AET): *“win basketball match.”*

Single-Level and Two-Level SVM Sentence Classifiers

For the sentence classification model, we proposed a simple single-level support vector machine (SVM) sentence classifier and a more sophisticated two-level hierarchical SVM sentence classifier.

The single-level SVM sentence classifier simply classifies all sentences as either positive (i.e., on-topic and event confirming) or negative (i.e., on-topic but non-event confirming, and off-topic).

The two-level SVM classifier attempts to distinguish sentences about current events from those about historical events as these sentences could otherwise confuse the single-level sentence classifier that is also responsible for distinguishing on- and off-topic sentences. The sentences about current events and historical events are known as *positive* and *historical sentences* respectively. For example, “the rejuvenated Celtics have won three straight since then and six straight at home overall” is a typical historical sentence, which is considered as “on topic” by the single-level classifier but hard to be identified as non-event confirming because the single level classifier is not trained to distinguish event confirming sentences from non-event confirming sentences. After applying two-level classification, this confusing case is easily solved.

Figure 5.5 shows the structure of the two-level SVM classifier. The first level classifier aims to detect all on-topic sentences, which include both positive and historical sentences. The second level classifier performs a refinement on the on-topic sentences by further classifying them as positive or historical.

Figure 5.5: 2-level SVM sentence classifier.

Sentence Classification Methods

We investigate various sentence retrieval strategies for AED, with a substantial focus on improving retrieval quality. In practice, the term weighting scheme used to represent a sentence vector has an enormous impact on the classification accuracy. The following methods using different term weighting schemes were compared in our experiments:

- Single-Level Classifier with {Standard TF, TFIDF, TFISF, TF+named entity features}
- Two-Level Classifier with {Standard TF, TFIDF, TFISF, TF+named entity features}

The standard TF scheme simply uses the raw frequency count of each term within a sentence. Another important factor to consider is the distribution of terms across a collection. Usually terms that are limited to a few sentences are useful for discriminating those sentences from the rest of the collection. This assumption leads to the introduction of ISF, called *inverse sentence frequency*. We also introduced the IDF, called *inverse document frequency*, at the sentence level to assume that terms appearing in a small number of documents are useful. The various term weighting schemes for a given feature x are summarized as follows:

$$\text{Standard TF} : f_{x,s}, \quad (\text{Eq. 5.3})$$

$$\text{TFIDF} : f_{x,s} \times \log\left(\frac{N}{N_x}\right), \quad (\text{Eq. 5.4})$$

$$\text{TFISF} : f_{x,s} \times \log\left(\frac{S}{S_x}\right), \quad (\text{Eq. 5.5})$$

where S is the total number of sentences in D , S_x is the number of sentences for which the word feature x exists in D and $f_{x,s}$ is the frequency of the word feature x in the sentence s . Our proposed weighting scheme, TF with named entities is simply standard TF appended with a team named entity feature and a score named entity feature denoting the frequencies of the two types of named entities respectively.

5.3.2 Document Classifier

Sentence retrieval is a very difficult problem [AWB03] by itself. This is because a single sentence contains neither enough information (curse-of-dimensionality) nor context to form a meaningful model. Thus, we proposed modeling the AET transition at the document resolution [HCL06b].

Document Classification Methods

We tried three different feature representation methods and one classifier combining strategy to train the AET classifier, as follows:

- CONTENT* : Entire news content as features.
- TITLE* : Title as features.
- 1SENT* : First sentence as features.
- VOTING* : Majority voting on the above three classifier outputs.

The *TITLE* and *1SENT* representations were inspired by the observation that human experts can usually decide if a news is a hit simply based on its first sentence and/or title. Moreover, the *TITLE* and *1SENT* representation of a news article may not always carry useful features, and the AED decision will have to fall back to the *CONTENT* representation. For example, the first sentence “*Signature Control Systems is off to a busy start in early 2006*” does not contain features really relevant to the “acquisition” event transition. *VOTING* was thus used as a simple and effective way to improve the overall accuracy.

Extracting Named Entities Types from Documents

In order to train a classifier on an AET using historically similar documents, it is very important to have the named entities replaced by named entity types [HCL06b]. For example, consider the following statements referring to two different “announces acquisition” events, with the named entities in boldface:

“*China*’s biggest computer maker, *Lenovo Group*, said on *Wednesday* it has acquired a majority stake in *IBM Corp*’s personal computer business in a deal worth a total value of *US\$1.75 billion (S\$2.86 billion)*, one of the biggest *Chinese* overseas acquisitions ever.”

“*SBC Communications* on *Monday* announced plans to acquire *AT&T* in a *\$16 billion* deal, a move designed to bolster *SBC*’s sales to enterprise customers nationwide and give it new national and global networks.”

In order for the two statements to be representative of the “post” state of the transition, it is better to replace the named entities with their name entity types, as follows:

GPE’s biggest computer maker, *ORGANIZATION*, said on *DATE* it has acquired a majority stake in *ORGANIZATION*’s personal computer business in a deal worth a

total value of **MONEY** (**ORGANIZATION MONEY**), one of the biggest **NATIONALITY** overseas acquisitions ever.

ORGANIZATION on **DATE** announced plans to acquire **ORGANIZATION** in a **MONEY** deal, a move designed to bolster **ORGANIZATION**'s sales to enterprise customers nationwide and give it new national and global networks.

Clearly, after the replacement, the two examples become more similar to one other, which invariably helps the classifier learn the event transition better.

5.4 Dataset

In this chapter, we introduce four datasets compiled by us as part of our work. Since AED is a new area defined by us, with no standard evaluation benchmark dataset, we created our own datasets for the experiments. They are: *Basket100* dataset, *Google Acquisition* Dataset, *Acquisition7* dataset, and *CNPC_Acquires_PKZ* dataset. For quality assurance purposes, each document in the four datasets was scrutinized and annotated by at least two people.

5.4.1 *Basket100* Dataset

In order to conduct FDA (Fisher Linear Discriminant Analysis) [Fis36] and KFDA (Kernel Fisher Discriminant Analysis) [MRW⁺99] visualization analysis on very sparse text data and evaluate the performance of sentence model for AED, we picked basketball matches as the topic of interest. The *Basket100* collection, which is based on the anticipatory event type “win basketball match”, comprises 100 documents returned by Google using the search terms “win basketball match”. The according ETG is shown in Figure 5.6.

Figure 5.6: The ETG for the *Basket100* dataset.

Distribution

In *Basket100*, 50 out of 100 documents are strongly relevant, i.e. only deals with current basketball matches, and the remaining 50 are irrelevant. However, many irrelevant documents contain historical basketball results and only a few of them are totally irrelevant

to “win basketball match”. Thus, we can say in *Basket100*, 93 out of 100 documents are relevant, i.e. deals with basketball matches, and the remaining 7 are totally irrelevant. The collection contains 2,340 sentences, comprising 4499 unique terms (words). The 2,340 sentences were manually classified into 3 categories:

- (i) positive class for “current basketball result”,
- (ii) negative class for “historical basketball results”,
- (iii) negative class for “irrelevant”.

Table 5.1 shows the summary statistics for the *Basket100* dataset.

Table 5.1: Class distribution of *Basket100* dataset.

<i>Document Class</i>	<i>Count</i>
Positive documents (class 1: current basketball result)	50
Negative documents (class 2: irrelevant)	50
Total	100
Positive sentences (class 1: current win basketball event)	189
Negative sentences (class 2: historical win basketball event)	117
Negative sentences (class 3: other irrelevant sentences)	2034
total	2340

Sparsity

Sparsity is a very important factor affecting sentence retrieval accuracy. The sparsity metric is computed as

$$Sparsity = \frac{n - \tilde{n}}{n}, \quad (\text{Eq. 5.6})$$

where $\tilde{n}$ is the average unique term count per sample. The Sparsity for *Basket100* is shown in Table 5.2.

From Tables 5.1 and 5.2, we can see that the *Basket100* dataset is unbalanced and very sparse, which is very typical of real world text data. Note that for each document or sentence, the average number of unique terms is much smaller than the average number of words, this is due to multiple occurrences of common terms. We also observe that sparsity at the sentence level is extremely bad, about 10 times as bad as the document level in terms of number of unique terms.

Table 5.2: Sparsity statistics for *Basket100* dataset.

<i>Average</i>	<i>per document</i>	<i>per sentence</i>
# terms	568.6	24.3
# unique terms	185.7	12.5
# unique terms in corpus	4499	4499
Sparsity	95.87%	99.81%

5.4.2 Google Acquisition Dataset

In order to learn an anticipatory event transition such as $t_{2,3}$ in Figure 5.2, we manually created the generic *Google Acquisition* dataset. The according ETG is shown in Figure 5.7. This dataset contains 346 as-it-happens news articles returned by Google News Alerts using

Figure 5.7: The ETG for the *Google Acquisition* dataset.

the keywords “announce acquisition”, the name of e_3 in Figure 5.2, during the two-month period from Dec 19, 2005 to Feb 19, 2006.

Each article in *Google Acquisition* is manually labeled as one of two possible events, i.e. “pre” or “post” $t_{2,3}$. Unfortunately, some articles can appear ambiguous even to a human expert. One general rule-of-thumb is to label the document based on overall context. For example, if the primary theme of an article revolves around the announcement, agreement, or completion of an acquisition, we label it as post- $t_{2,3}$; otherwise, it is labeled as pre- $t_{2,3}$. We note that the latter case could also include irrelevant documents completely unrelated to acquisition.

In order to ensure consistency in annotation, the following set of guidelines was established:

- If an article mainly describes an acquisition or related activities including announcement, agreement, or completion of an acquisition, we annotate it as post- $t_{2,3}$ or “positive” for $t_{2,3}$.

- Any news article alluding to a previously announced acquisition is annotated as “positive” for $t_{2,3}$, provided it primarily describes an acquisition event. Thus, a case like “Company A delays closing of acquisition of Company B” is considered “positive” for $t_{2,3}$.
- If an article talks about something that is completely unrelated to acquisition events, it is annotated as “negative” for $t_{2,3}$.
- If an off-topic article contains just a few statements about acquisition, and which may only be mentioned lightly, or in a historical context (e.g. containing a single sentence without any supporting context), we annotate it as “negative” for $t_{2,3}$.
- If an article is a cento or hub of many breaking news (for example, a monthly roundup of the nation’s breaking business news), of which one talks about an acquisition, we annotate it as “negative” with respect to $t_{2,3}$, since it contains marginally more information than the previous off-topic case.
- If a news article writes about rumors of some pending acquisition that has not been officially announced, we annotate it as “negative” for $t_{2,3}$.
- Any news article mentioning an expected announcement of acquisition will be annotated as “negative” for $t_{2,3}$.

Based on the above criteria, we annotated 178 documents as positive and 168 as negative, which means that Google News Alerts returned 168 or almost half false alarms (48.6%) for the subscribed keywords “announce acquisition”. This is a typical result from the simplistic keyword based news alert system.

5.4.3 *Acquisition7* dataset

We created the third dataset, *acquisition7*, which covers seven recent acquisition topics as the test data for our proposed online AED solution, using the same ETG of the *Google Acquisition* dataset (shown in Figure 5.7). Each acquisition news topic in the *acquisition7* collection comprises 20 plus news articles returned by Google News, approximately half of each (10) were reported before and after $t_{2,3}$. The major difference between this dataset and the *Google Acquisition* dataset is that each document is not generic but instead tied to

a particular acquisition topic. Further, there are no irrelevant documents in this dataset; a document occurs either before or after the $t_{2,3}$ for a given topic.

The 7 acquisition news topics are listed in Table 5.3, where $t(e_3)$ refers to the true time of occurrence for e_3 in Figure 5.2. The annotation of *Acquisition7* dataset follows the same criteria as defined in Section 5.4.2.

<i>Acquisition Topics</i>	$t(e_3)$
Adobe acquires Macromedia	Apr 18, 2005
State-owned Chinese oil company CNPC acquires PetroKazakhstan (PKZ)	Oct 26, 2005
eBay acquires Skype	Sep 12, 2005
Lenovo acquires IBM PC Division	Dec 08, 2004
Oracle acquires PeopleSoft	Dec 13, 2004
Oracle acquires Siebel	Sep 12, 2005
SBC acquires AT&T	Jan 31, 2005

Table 5.3: The make up of *Acquisition7* dataset.

5.4.4 *CNPC_Acquires_PKZ* dataset

The *CNPC_Acquires_PKZ* dataset was singled out from the *Acquisition7* dataset as a case study for understanding acquisition events. It contains 61 news articles entirely about the acquisition topic “State-owned Chinese oil company CNPC acquires PetroKazakhstan (PKZ)”, and was created to help illustrate the process of manually creating the entire *Acquisition7* dataset. The detailed description of *CNPC_Acquires_PKZ* dataset is summarized as follows.

- **Topic title:** State-owned Chinese oil company CNPC acquires PetroKazakhstan (PKZ)
- **Data source:** documents returned by Google News with the search keyword “China Oil purchase PetroKazakhstan” and “CNPC buy PetroKazakhstan”.
- **Total number of news articles:** 61
- **Number of positive news articles:** 29

- **Number of negative news articles:** 32
- **XML format of news articles:**

```

<DOC>                                //start of XML format

<DOCNO>1</DOCNO>                     //numerical index

<DOCTYPE>positive</DOCTYPE> // {positive, negative}

<SOURCE>Bloomberg</SOURCE> //verify the reliability of this news
                             documents
<TITLE>...</TITLE>                //extracted from the news content

<DATE>...</DATE>                   //extracted from the news content

<TEXT>...</TEXT>                   //main part of news content

</DOC>

//end of XML format

```

- **Ordering of documents:** Each news article is assigned a DOCNO ID and ordered from earliest to latest (most recent) as follows:

$$(\textit{oldest})\ 82, 81, 80, \dots, 53, 52, 51, 1, 2, 3, \dots, 27, 28, 29\ (\textit{newest}) \quad (\text{Eq. 5.7})$$

where the first story after the transition has $DOCNO = 1$ which means that the set of positive documents has $DOCNO \geq 1$.

- **Anticipatory event transition:** Chinese oil company CNPC announced the acquisition of PetroKazakhstan (PKZ) ($t_{2,3}$ in Figure 5.2).
- **Positive class:**
 - (i) DOCNO 1-24: directly report on the announcement/completion of CNPC's acquisition of PKZ.
 - (ii) DOCNO 25-29: describe clashes between CNPC and Nelson.
- **Negative class:**
 - (i) Off-topic Documents
 - Talisman buy Paladin (another two oil firms): DOCNO 55
 - Rise of Canadian Stocks: DOCNO 66

Other Issues on Russian Oil firms: DOCNO 67, 74

Occidental buy Vintage (another two oil firms): DOCNO 70, 72

Other Issues on Chinese Oil firms: DOCNO 77

- (ii) On-topic documents before the true happening of the anticipatory event transition

DOCNO 51-54, 56-65, 68-69, 73, 75-76, 78-82

- **First story of *CNPC_Acquires_PKZ* dataset:**

Date: Sept. 30, 2005

Title of first story: Lukoil to Buy Kazakh Oil Producer Nelson for \$2 Bln (Update7)

DOCNO: 82

- **First story after anticipatory event transition:**

Date: Oct. 26, 2005

Title of first story: China Wins PetroKazakhstan Bid, Repels India, Russia (Update1)

DOCNO: 1

5.5 Exploratory Data Analysis

To gain more insight into the data, we also performed two exploratory data analysis experiments on the *Basket100* dataset and the *CNPC_Acquires_PKZ* dataset respectively.

5.5.1 FDA and KFDA Visualization Analysis

We conducted FDA and KFDA analysis using the *Basket100* dataset and concluded that shorter instance vectors (like sentence) can be visualized well using FDA and KFDA, which means linear classification on these subspace should also work well. The standard information retrieval measures, Precision and Recall, are adopted to evaluate the performance of the various visualization methods.

We found that most visualization methods seem to work well for text data because text data is very sparse. It is relatively easy to generate a direction to discriminate training data. But most separating hyperplanes, including the linear discriminant, tend to over-fit the training data. To achieve more objectivity, 10-fold cross-validation was performed to evaluate the models generated by FDA/KFDA on our test dataset.

Preprocessing

Incremental Term Frequency (ITF) weighting combined with named entities [HCL06a] was used, and documents/sentences cosine similarity was adopted as the kernel function. We follow the same way of applying named entities in the sentence model for AED (see Section 5.3.1) by isolating named entities to enhance the feature vector of sentences.

Analyzing Tool

The popular free Statistical Pattern Recognition Toolbox [Spr] for Matlab was used to analyze data. This toolbox implements a selection of statistical pattern recognition methods including FDA (`fld.m/fldqp.m`) and KFDA (`kfd.m`).

- **`fld.m/fldqp.m`** The input is a set $\mathbf{D} = (\mathbf{d}_1, l_1), \dots, (\mathbf{d}_N, l_N)$ of binary labeled $l_i \in 1, 2$ training vectors $\mathbf{d}_i \in \mathbb{R}^n$. STPRtool contains two implementations of FDA, one based on classical matrix inversion (implemented in `fld.m`), and the other one based on a quadratic optimization approach (implemented in `fldqp.m`). The QP solver `quadprog` of the Matlab Optimization Toolbox is used in the latter implementation. The optimization implementation is quite useful since for very high dimensional text data, matrix inversion is hard to compute. The first FDA projection $\mathbf{d}'_1$ is generated using

$$\mathbf{d}'_1 = \mathbf{D}\mathbf{w} + b,$$

where $\mathbf{w}$ is the transformation vector and b is a scalar constant. According to [oaD05], generating the second FDA projection vector in a two-class problem is not so trivial. This is due to the fact that the rank of the $\mathbf{\Sigma}_B$ matrix (between classes scatter matrix) is 1 (or $k - 1$ for a k class problem) and $\mathbf{\Sigma}_W$ (within classes scatter matrix) may be ill-conditioned. We adopted the method in [oaD05] by using pseudoinverse for $\mathbf{\Sigma}_W$ and perturbing the $\mathbf{\Sigma}_W$ values to generate the second FDA projection vector. A similar procedure was used to generate the second KFDA projection vector.

Example:

```
trn = load('riply_trn'); % load training data
model = fldqp(trn); % compute FLD
```

```

figure;
ppatterns(trn); pline(model); % plot data and solution
tst = load('riply_tst'); % load testing data
ypred = linclass(tst.X,model); % classify testing data
cerror(ypred,tst.y) % compute testing error

ans = 0.1080

```

- **kfd.m** The input is the same as in the previous case. The **kfd.m** matlab script is the non-linear extension of the linear **fld.m**.

The first KFDA projection is generated using

$$\mathbf{d}'_1 = \sum_{i=1}^N \alpha_i \Phi(\mathbf{d}_i, \mathbf{D}) + b \quad (\text{Eq. 5.8})$$

where Φ is defined a kernel function operating on a set of vector samples in $\mathbf{D}$.

Example:

```

trn = load('riply_trn'); % load training data
options.ker = 'rbf'; % use RBF kernel
options.arg = 1; % kernel argument
options.C = 10; % regularization of linear SVM
options.mu = 0.001; % regularization of KFD criterion
model = kfd(trn, options) % KFD training

% visualization
figure; ppatterns(trn); psvm(model);

% evaluation on testing data
tst = load('riply_tst'); ypred = svmclass( tst.X, model ); cerror(
ypred, tst.y ) ans = 0.0960

```

Fisher Linear Discriminant Analysis

- Document-level dataset, 2 classes

As shown in section 5.4.1, the dimension of the document vector is 4499. The sum Σ_W is thus a 4499×4499 matrix, and almost singular. It is therefore impractical to compute the inverse matrix for Σ_W . So, quadratic programming (fldqp.m, see section 5.5.1) is used for FDA.

FDA visualization for the document-level dataset is shown in Figure 5.8. From Figure 5.8, we can see that the two classes are nicely separated. After 10-fold cross-validation,

Figure 5.8: FDA for document-level dataset

the performance is shown in Table 5.4. With a 0.2 average precision and a 0.2 average recall for both classes, FDA overfits since it shows good separation in the training dataset but poor test classification performance. The reason is that like most linear discriminants, FDA also uses Euclidean distance (dot product) between two vectors as the similarity metric. However, in general, the dot product between two vectors is of no use as a similarity metric for very sparse data, since it is too sensitive to the absolute magnitude of the various dimensions.

Table 5.4: 10-fold cross-validation results for FDA on document-level dataset.

<i>Class</i>	<i>Precision</i>	<i>Recall</i>
positive	0.200	0.200
negative	0.200	0.200

- Sentence-level dataset, 3 classes

The FDA visualizations for sentence-level 3-class dataset are shown in Figures 5.9, 5.10 and 5.11 respectively, and the corresponding 10-fold cross-validation classification results are shown in Tables 5.5, 5.6, and 5.7.

Figure 5.9: FDA for sentence-level dataset (class 1 vs. class 2)

From Figures 5.9 to 5.11, we can make two intuitive observations:

- It is harder to distinguish between class 1 and class 2 (shown in Figure 5.9, as there are more overlapping points between them than the other two pairs. That is reasonable since class 1 (current win event) and class 2 (historical win event) are more similar to each other. However, most of them are still separable, because

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

Table 5.5: 10-fold cross-validation results for FDA on sentence-level dataset (class 1 vs. class 2)

<i>Class</i>	<i>Precision</i>	<i>Recall</i>
positive	1	0.545
negative	0.688	1

Figure 5.10: FDA for sentence-level dataset (class 1 vs. class 3)

Table 5.6: 10-fold cross-validation results for FDA on sentence-level dataset (class 1 vs. class 3)

<i>Class</i>	<i>Precision</i>	<i>Recall</i>
positive	0.625	0.455
negative	0.962	0.980

as mentioned in Section 5.5.1, we added some domain-based named entities as new features to help in the classification; otherwise, classification results could have been worse.

- (ii) Although the test results are not decent, the heavy “overfitting” that happened

Figure 5.11: FDA for sentence-level dataset (class 2 vs. class 3)

Table 5.7: 10-fold cross-validation results for FDA on sentence-level dataset (class 2 vs. class 3)

<i>Class</i>	<i>Precision</i>	<i>Recall</i>
positive	0.600	0.545
negative	0.968	0.974

in the document-level dataset did not happen here. The reason is that for document-level dataset, as shown in Section 5.4.1, the average number of features per document is 185.7, which is far larger than the average number of features for a sentence (12.5). Therefore, a test document has a higher probability to be influenced by the new features. We believe there exists a trade-off between the average length of instances (document or sentence) and the influence by the magnitude of new features. When the average length of instances is very small (like sentence), the instance will be influenced less by new features. However, when the average length of instance is big enough to reduce the effect of data sparsity, the probability of the occurrence of new features will drop significantly. In general, for very sparse text data, sentence-level instance vectors are more

suitable inputs for linear discrimination.

Kernel Discriminant Analysis with Cosine Similarity Kernel

The *kernel* represents the similarity between two objects. For very sparse text data, the magnitude of instance vectors is of no use as a similarity metric, since it is too sensitive to the absolute magnitudes of the various dimensions. However, the normalized dot product between vectors has a useful and intuitive interpretation: it computes the cosine of the angle between two vectors. When two documents are identical they have a cosine similarity of one; when they are orthogonal (share no common terms) they will yield a cosine similarity value of zero. We thus use the cosine similarity as the kernel for KFDA on the *Basket100* dataset.

- Document-level dataset, 2 classes

The KFDA visualization with cosine similarity kernels for document-level dataset is shown in Figure 5.12, and the corresponding 10-fold cross-validation results is shown in Table 5.8.

Figure 5.12: KFDA (cosine similarity matrix) for document-level dataset

Table 5.8: 10-fold cross-validation results for KFDA (cosine similarity matrix) on document-level dataset

<i>Class</i>	<i>Precision</i>	<i>Recall</i>
positive	0.750	0.600
negative	0.667	0.800

Amazingly, KFDA with cosine similarity matrix was the only method that could be applied to the document-level dataset, since all other FDA/KFDA methods failed to give satisfactory results. Although its performance (precision 0.750 and recall 0.600 on positive class) still needs improvement.

- Sentence-level dataset, 3 classes

Due to the fact that it is hard to compute the inverse matrix for very high dimensional matrix (which is nearly singular), and to ensure a fairer comparison, only class 1 and class 2 of the sentence-level dataset are considered here. The KFDA visualizations with cosine similarity kernels for the sentence-level dataset (class 1 and class 2) are shown in Figure 5.13, and the corresponding 10-fold cross-validation test results are shown in Table 5.9.

Table 5.9: 10-fold cross-validation results for KFDA (cosine similarity matrix) on sentence-level dataset (class 1 vs. class 2)

<i>Class</i>	<i>Precision</i>	<i>Recall</i>
positive	0.889	0.727
negative	0.769	0.909

The performance of KFDA with cosine similarity kernels is relatively good. Actually it performed the best among all methods used in our exploratory data analysis experiments. The result simply validated the usefulness of the cosine similarity metric as a kernel function for KFDA.

Figure 5.13: KFDA (cosine similarity matrix) for sentence-level dataset (class 1 vs. class 2)

Discussion

We have presented a series of experiments applying different visualization methods on the *Basketball100* dataset. To gain a better understanding of the over-fitting problem, and to find a good approach to classify very sparse text data, we ran all experiments on both the document-level and sentence-level datasets. As extensions to the baseline FDA and KFDA, we also introduced cosine similarity matrix as a new kernel matrix for KFDA on text data.

Two important findings are as follows:

- (i) Cosine similarity is a good similarity metric for very sparse text data, compared to linear dot product metric and other popular kernel functions.
- (ii) Shorter instance vectors (like sentence-level) can be visualized well using FDA and KFDA, which also leads to better classification accuracies.

5.5.2 Analysis of Acquisition Dataset

We choose the *CNPC_Acquires_PKZ* dataset, as a typical representative of the *Acquisition7* dataset, for our analysis. The 10-fold cross-validation results is conducted using the CONTENT classification method.

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

Binary document vector representation is used here and the approach of applying named entities in the document model for AED (see Section 5.3.2) by replacing named entities with named entities types is also employed. SVM-light [Svm] is used here as the classification method. Figure 5.14 shows the experimental results for document retrieval accuracy with 10-fold cross-validation. We see from Figure 5.14 that except for some in-sample validation

Figure 5.14: 10-fold cross-validation results on the *CNPC_Acquires_PKZ* dataset.

errors in the 1st, 7th and 9th fold, both 100% precision and recall have been achieved for the positive class in the other folds.

The overall performance is given by Table 5.10 and Table 5.11: From Table 5.10, we see

Table 5.10: Average classification results.

	<i>Predicted positive class</i>	<i>Predicted negative class</i>
True positive documents	27	2
True negative documents	3	29

that a total of 5 errors happened in the 10-fold cross-validation; 2 positive documents (over all 29 positive documents) were missed and 3 negative documents (over all 32 documents)

were mis-classified. Shown in Table 5.11, we achieved a decent in-sample accuracy (above 90% of in-sample precision and recall for both positive and negative classes).

Table 5.11: Average evaluation for both positive and negative classes.

	<i>Precision</i>	<i>Recall</i>	<i>F Measure</i>
Positive class	0.9	0.931	0.915
Negative class	0.935	0.906	0.92

Based on the above results, two important questions must be raised:

- (i) What are the most discriminatory features for anticipatory event transition?
- (ii) How did the 5 errors happen?

We investigate these issues in the following subsections.

Discriminatory features for AED

We use Odds Ratio (OR) Feature Selection [Mla98] method to discover the most discriminatory features for AED. OR is a one-sided metric that ranks features according to their indication of membership for a category only (i.e. the target category).

In text classification tasks, several variants of OR formulas have been proposed and evaluated in [Mla98]. The exponential representation delivered comparable effectiveness compared to (if not better than) the rest. Hence, we adopt the exponential OR which is defined as

$$OR(x|z) = e^{P(x|z) - P(x|\bar{z})}, \quad (\text{Eq. 5.9})$$

where $P(x|z)$ and $P(x|\bar{z})$ are the conditional probabilities of feature x appearing in class c and class $\bar{z}$ (not in z) respectively, and $OR(x|z)$ is the discriminative score of feature x in class z . A higher value of $OR(x|z)$ means that the feature x is more discriminative w.r.t. class z . The definition of $P(x|z)$ is dependent on the document feature vector representation [MN98]. For a given feature x_j , if a binary feature vector is used to represent a document, $P(x_j|z)$ is defined by Eq. 5.10; if term frequencies are used in the document feature vector representation, $P(x_j|z)$ is given by Eq. 5.11.

$$P(x_j|z) = \frac{1 + \sum_{d_i \in z} w_{ji}}{2 + N(z)} \quad (\text{Eq. 5.10})$$

$$P(x_j|z) = \frac{1 + \sum_{d_i \in z} w_{ji}}{n + \sum_{l=1}^n \sum_{d_i \in z} w_{li}} \quad (\text{Eq. 5.11})$$

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

Based on the Odds Ratio Feature Selection, the top 50 most discriminatory features are given as follows. Note that some strange words appear in the list due to tokenization errors.

after	aiyar	GPE	approval	approved
are	as	asked	attempt	based
been	biggest	MONEY	by	calgary
called	canadian	central	DATE	conclude
LOCATION	court	decision	declined	delayed
acquires	details	disputed	distributed	dmitry
dolgov	ease	economy	enabling	ending
event	fastest	from	fuel	giving
globe	has	inc	included	NATIONALITY
ORGANIZATION	it	its	joint	acquire

From the above top 50 discriminatory features, we see that AED is a very special classification problem because it is impossible for us to guess what are the exact features that will decide the final classification. Heuristically the feature “after” should not be a discriminated features in a typical classification problem because it is usually a stopword; however here it is the most discriminative feature for the “announce acquisition” transition with the removal of only a few articles since we would like to preserve all the possible time-sensitive structures. Normally the feature dimension of news ranges in the tens of thousands, and the SVM decision hyperplane cannot be interpreted intuitively. Therefore, we concluded that for AED classification, a good conceptual model and a reasonable training method are far more useful than considering a few important features.

Error Analysis

There were altogether 5 errors; 2 positive documents were missed and 3 negative documents were wrongly classified positive (we will only examine the positive class from now on). An overall summarization of these 5 errors is given in Table 5.5.2.

In Table 5.5.2, time sequence refers to the order of a document in its true class. Thus, position 1 indicates that the document is the first reported document within its class. Although the classifier misclassified 2 positive news documents, it was able to identified the

Table 5.12: Overall summarization of 5 errors in *CNPC_Acquires_PKZ* dataset.

<i>Error Index</i>	<i>Error Type</i>	<i>True Class Label</i>	<i>DOCNO</i>	<i>Time Sequence</i>
1	miss	positive	20	20
2	miss	positive	26	26
3	false alarm	negative	53	30
4	false alarm	negative	75	8
5	false alarm	negative	76	7

remaining 27 positive news documents, among which includes correctly classified positive news documents predating the missed ones. This result is significant because actual on-line AED is a winner-take-all-later-ones system where all subsequent news articles automatically defaults to positive once the first positive story has been identified by the AED system. Thus, in practice, the 2 mis-classified positive news document would have been defaulted to positive simply by virtue of an earlier identified positive document.

In the following we analyze each error individually.

- **Error 1:**

DOCNO: 20

Title of news article: China eyes share of Kazakh oil

Prediction score returned by SVM-light: -0.40077688 (will be classified into positive class if predicted a positive value and vice versa)

Average positive prediction score: 0.560783975

Average negative prediction score: -0.391534267

Analysis: Two sentences (called positive sentences) can indicate the true occurrence of the anticipatory news event “Chinese oil company CNPC completed the acquisition of PetroKazakhstan (PKZ)”. The first one is “China’s biggest foreign energy takeover ever took a giant step forward yesterday as a Canadian judge approved a \$4.18 billion bid for control of Petrokazakhstan’s oil fields, following a bidding war with Russian oil giant Lukoil” and the second one is “(CNPC) finally got approval to buy Calgary-based Petrokazakhstan, despite Lukoil’s rival offer”. However, as shown above, the prediction scores returned by SVM-light is even lower than the average negative prediction score. The main reason is that in the real world most newspapers used the

approval of the Canadian court as the indicator of completed acquisition. However, this document just mentioned “a giant step” towards the Canadian court’s approval in the first positive sentence. Although in the second positive sentence “finally got approval to buy” is mentioned and the word “approval” seems like a more discriminative feature to indicate the happening of event, unfortunately, within the 25 documents containing the word “approval”, 9 of them are from the negative class. Therefore, the second positive sentences does not give much support to correct classification. An interesting thing is that if we build bigram features to enhance the whole feature space, the bigram features like “got approval”, “approval to” come from the clause “finally got approval to buy” only exist in positive documents. Thus, by isolating bi-gram features we might not have missed this document.

- **Error 2:**

DOCNO: 26

Title of news article: China Petroleum Fights Lukoil for Kazakh Oil Assets (Update1)

Prediction score returned by SVM-light: -0.08970224

Average positive prediction score: 0.6383461675

Average negative prediction score: -0.2019495367

Analysis: Only one positive sentence, “CNPC yesterday completed its \$4.2 billion takeover of PetroKazakhstan Inc., which pumps 12 percent of Kazakhstan’s oil”, exists. But now the situation is even worse although this document is just marginally negative based on the prediction score. Because the most important keyword “completed” exists in 9 documents, 7 of which are in negative class! The only hope to avoid this type of misses is to combine “completed” with “takeover” as a strong bi-gram feature. Unfortunately, these two words are isolated by some noise words in the original sentence. A possible solution is to train Natural Language Processing (NLP) patterns. In this case, using a basic single slot NLP pattern should suffice.

- **Error 3:**

DOCNO: 53

Title of news article: LUKOIL says has Kazakh blessing to buy Nelson

Prediction score returned by SVM-light: 0.31191982

Average positive prediction score: 0.4537951925

Average negative prediction score: -0.0676994765

Analysis: This document actually discussed “LUKOIL’s acquisition of Nelson”, but unfortunately, it also mentioned “CNPC’s takeover of Canada’s PetroKazakhstan”. Since we did the classification at the document level, it is not surprising that the classifier got confused about “who bought who” and gave this document a high positive prediction score. This problem could be solved easily at the sentence-level classification as shown in our previous work [HCL06a].

- **Error 4:**

DOCNO: 75

Title of news article: China’s CNPC optimistic on PetroKazakhstan deal despite Lukoil suit - source

Prediction score returned by SVM-light: 0.60885957

Average positive prediction score: 0.6383461675

Average negative prediction score: -0.2019495367

Analysis: There are lots of unexpected clauses in this document: “before CNPC took over PetroKazakhstan”, “Should a Canadian court approve the CNPC purchase” and “CNPC agreed to buy PetroKazakhstan” etc. It is really hard to fix this misclassification using anything short of a real human. Even if we use a sophisticated NLP system, it is hard to train a system to understand the different language tenses. A theoretical solution is to store all possible language patterns in the memory as a look-up table, however that is not currently feasible.

- **Error 5:**

DOCNO: 76

Title of news article: Lukoil objects to CNPC purchase of Canada’s PetroKazakhstan

Prediction score returned by SVM-light: 0.65001634

Average positive prediction score: 0.6383461675

Average negative prediction score: -0.2019495367

Analysis: As mentioned in section 4.1, the first document of the anticipatory news event happened on Oct. 26, 2005. On that day the Canadian court approved the

CNPC’s acquisition of PetroKazakhstan. However, the funny thing is that a document reported on Oct. 11, 2005 claimed that “Russia’s Lukoil Overseas Kumkol BV will oppose court approval of China National Petroleum Corporation (CNPC)’s purchase of the Canadian oil firm”. The word “will” is really a trap even for human beings to grasp the true transition correctly. Therefore, SVM-light gave this document a very high (above average) positive prediction score. Currently we do not have a good solution to solve this mis-classified case.

Discussion

In spite of the 5 errors reported in the experiment, compared to previous results reported for online topic detection [ALJ00, KA04], we have actually achieved an impressive feat. The main reason is that we built a clear anticipatory event transition model for events of the same *topic type*, which is shown in Figure 5.2 on the *topic type* “acquisition”. On the other hand, previous online topic detection approaches suffered from lack of training data, and they compared the current document on hand with all previous seen documents irregardless of *topic type*. The intuitive idea is to build an on-line classification system but in practice their systems got confused in the sea of language nuances and semantics. Through the careful modeling of news event data via training, we found that our anticipatory event transition model is able to distinguish documents of the same *topic type* according to the transition modeled. Our novel idea really contributed in improving the accuracy of topic detection.

5.6 Experiments on AED

The data analyses performed in section 5.5 showed that our datasets are separable, and can potentially be classified using a conceptual model such as AED. In this chapter, we conduct a series of experiments on the sentence level and Document level AED Models. Experimental results validated the applicability and practicality of our proposed AED models.

5.6.1 Experimental Setup

Version 2 of the open source Lucene software was used to tokenize the news text content, remove stopwords, and generate the document-word vector. In order to preserve time-sensitive past/present/future tenses of verbs, no stemming was done other than the removal of a few articles.

We used a normalized (unit length) binary document vector representation because we observed that co-occurrences of terms are far more important than the raw term frequency and inverse document frequency for AED. The normalized binary document-word training or test vectors are then fed into SVM-light [Svm] for training and classification, respectively.

Due to the way AED is defined, there will clearly be a lot more negative sentences compared to positive sentences, resulting in what we call a highly unbalanced training dataset. This could easily lead to useless trained classifiers that classify all sentences as negative. SVM Cost factors [MBJ99], which assign a penalty to wrong classifications, can be used to counter this problem. The cost-factors are typically chosen so that the potential total cost of the false positives C_+ equals the potential total cost of the false negatives C_- , obeying the ratio:

$$\frac{C_+}{C_-} = \frac{\# \text{ negative training examples}}{\# \text{ positive training examples}} \quad (\text{Eq. 5.12})$$

Therefore, SVM Cost factors were employed in our experiments to offset any discrepancies in numbers between the positive and negative documents.

All algorithms were implemented in Java. A single machine was utilized since speed (computational efficiency) was not considered as an important factor of our proposed AED evaluation.

In our experiments, we use BBN's Identifinder [BSW99] to identify 24 types of named entities, including *Animal*, *Contact info*, *Disease*, *Event*, *Facility*, *Game*, *Geo-political entities*, *Language*, *Law*, *Location*, *Nationality*, *Organization*, *Person*, *Plant*, *Product*, *Substance*, *Work of art*, *Date*, *Time*, *Cardinal*, *Money*, *Ordinal*, *Percentages*, and *Quantity*. Extracted named entities are then replaced in line by one of the 24 named entity types.

5.6.2 Sentence AED Model

The sentence model was one of the first solutions proposed by us to solve the AED problem during a pilot study. The gathering of training documents can be viewed as a form of user preference, albeit an extremely laborious process. Two classifiers using various term weighting schemes of the sentence model were applied to the *Basket100* dataset, with the goal of detecting a winning basketball event.

Figure 5.15: 1-level SVM classification with 10-fold cross-validation.

Single-Level SVM Sentence Classifier

Figure 5.15 shows the classification results of the single-level SVM using various term weighting schemes. We see that the sentence classifier using our proposed weighting scheme yielded the best F-Measure, compared to all other methods. Although this method yielded a very respectable 0.75 for precision, its recall at 0.62 still leaves much to be desired. The low recall is due to large number of false negatives.

The other methods fared significantly worse. TFISF performed worse than TF alone, probably due to the fact that there were too many negative (including historical) sentences biasing the weight of discriminatory terms involved with the positive sentences. Further, we also need to take into account that for single-level classification, both positive and historical winnings contain similar terms, but are labeled as positive and negative, respectively. TFIDF performed the worst, due to the large discrepancy between a term at the sentence and document levels.

Two-Level SVM Sentence Classifier

Figures 5.16-5.18 show the experimental results of sentence classification for the two-level classification method using different term weighting schemes. Since the first level classifier

Figure 5.16: Results for the first level of the 2-level SVM classifier with 10-fold cross-validation.

Figure 5.17: Results for the second level of the 2-level SVM classifier with 10-fold cross-validation.

is only responsible for distinguishing on-topic sentences from off-topic ones, its performance was measured based on the on-topic sentences, as opposed to positive sentences.

Figures 5.16-5.17 show that precision values at both levels were not affected much by the different weighting schemes. This confirmed our previous suspicion that the similar-

Figure 5.18: Overall results of 2-level SVM classifiers with 10-fold cross-validation.

ity between positive and historical sentences was a large contributing factor to the low precision when inverse document and sentence frequencies come into play. The overall performances of the classifier is shown in Figure 5.18, achieving a precision/recall of 0.68 and 0.72 respectively.

Note that due to the lack of training samples at the second level, the classifier at level two may not accurately categorize a false positive sample. That is, on average, half of the false positive samples from level one will be classified into positive and the other half into historical. Thus, the first level classifier is crucial to the success of the second level classifier, and thus the overall accuracy.

Discussions

Figure 5.19 shows the comparison of F1-Measures between single-level and two-level classification under different term weighting schemes. It can be seen that a two-level sentence classifier based on the simple TF and named entity weighting scheme gives the best overall performance.

Numbers aside, the practical implications for AED is simply this: if high precision is desired, go with the 1-level classifier. This means that if a subscriber is waken up at night, it is probably due to a valid news event. However, he loses by missing some important

Figure 5.19: Comparison of F1-Measures between 1-level and 2-level classification

events due to the low recall. On the other hand, if he is willing to put up with 8% less precision (i.e. waken up at night by irrelevant alerts), he stands to catch 10% more of the actual alerts by going with the 2-level classifier. Thus, each approach has its pros and cons, and the ultimate choice is best left to the news alert subscriber.

Sentence classification is not new, but up till now, results previously reported for sentence classification have been very dismal [AWB03]. Thus, one contribution of our work is to demonstrate that good sentence classification performance (70% for both precision and recall) is attainable if the problem domain is well defined, as in AED. Nevertheless, we believe that a document based classification approach could possibly achieve much better results as a document contains significantly more information than a sentence (as much as 15 times as much unique terms as shown in table 5.2).

5.6.3 Experiments on Document AED Model

After the limited success with Sentence AED Model, we perform AED at the document level, using a better-tuned document classifier.

AED via Cosine Similarity on *Acquisition7*

As a baseline, we evaluated AED performance using simple cosine similarity on the *Acquisition7* dataset. Standard TFIDF document representation was used with the following variations [KA04]: all terms (All Terms), all terms without named entities (No NE), and name entities only (NE only).

Each incoming news article is compared to *all* existing news articles (assumed to be negative or pre-transition) from all 7 topics. If the cosine similarity between this news and its nearest neighbor falls below a threshold, the incoming news is considered to have consummated the transition; otherwise it is classified as a negative news.

The similarity approach generated largely false alarms and misses, except for one accurate alarm, for various values of similarity thresholds. Figure 5.20 thus shows the ratio of misses to false alarms for all 3 vector representations versus a gradually increasing similarity threshold. From Figure 5.20, we observe that starting from a low similarity threshold, the system was initially very strict (incoming news must be significantly different, i.e. has low cosine similarity compared to all existing news), resulting in high percentage of misses. As the similarity threshold is gradually increased, the system was able to detect some news, but almost all prematurely as false-alarms, except for one accurate alarm detected by the “no NE” representation. Our results clearly show that similarity based approaches, which does not use a conceptual model, are too simplistic to detect a transition leading to a user-desired AE.

Event-conditioned Novelty Detection for *Acquisition7*

Event-conditioned 1NN novelty detection [YZCJ02], essentially a topic-constrained cosine similarity approach, first classifies a news into a known topic before applying cosine similarity comparison between it and its nearest neighbors. To model this approach, we applied cosine similarity AED to all news within the same topic to obtain the results listed in Table 5.13. From Table 5.13, it is obvious that event-conditioned novelty detection failed the AED task miserably as it generated all but one false alarms. This again shows that similarity approaches work poorly even when documents are constrained with the same topic.

Figure 5.20: AED using cosine similarity on Acquisition7 Dataset.

Table 5.13: Event-conditioned novelty detection on *Detection of Anticipatory Event Transition* for *Acquisition7* dataset.

<i>Alarms</i>	<i>Accurate</i>	<i>Delayed</i>	<i>False</i>	<i>Miss</i>
Adobe acquire Macromedia			✓	
CNPC acquires PetroKazakhstan (PKZ)			✓	
eBay acquire Skype	✓			
Lenovo acquire IBM PC Division			✓	
Oracle acquire PeopleSoft			✓	
Oracle acquire Siebel			✓	
SBC acquire AT&T			✓	

Validating the Google Acquisition Dataset

In Sections 5.6.3 and 5.6.3 we have showed empirically that the cosine similarity [KA04] and event-conditioned novelty detection [YZCJ02] basically failed in the AED task. One of the reason could be the lack of acquisition training data, since we could not use the 7 test topics to train for AED. Thus, we would like to use the generic event-agnostic *Google Acquisition* dataset to train a $t_{2,3}$ model, which hopefully can generalize to detect the majority of real-world $t_{2,3}$. However, before using the dataset, we would like some sort of quality assurance that it is indeed a reasonable dataset for modeling $t_{2,3}$.

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

To this end, we conducted a two-fold cross-validation using the four proposed text classification methods of Section 5.3.2 on the *Google Acquisition* dataset. The dataset is first split along the timeline into two equal parts: 1) news articles dating from Dec 19, 2005 to Jan 19, 2006, and 2) news articles dating from Jan 20, 2006 to Feb 19, 2006. One part was used for training and the other part used for testing the trained models and vice-versa. The test performance is summarized in Figure 5.21 and Table 5.14.

Figure 5.21: Average test results of the four text classification methods applied to *Google Acquisition*

Table 5.14: Average test results on *Google Acquisition*.

	<i>CONTENT</i>	<i>TITLE</i>	<i>1SEN</i>	<i>VOTING</i>
average # false alarms	22.5	15.5	17	13.5
average # misses	9	24.5	15	10
average precision	0.7847	0.8110	0.8172	0.8571
average recall	0.9011	0.7308	0.8352	0.8901
average F-Measure	0.8389	0.7688	0.8261	0.8733

From Figure 5.21 and Table 5.14, we see that the VOTING strategy is the overall best performer with the least number of false alarms, while the CONTENT method gives a slightly higher recall at the expense of almost twice as many false alarms. The main problem with the CONTENT method is that it is easily affected by a few transition-alluding sentences in negative documents. One such sentence is as follows: “Additionally, Magazine

Acquisition announced that Morgan Stanley Real Estate and Onex Real Estate will be partnering with Sawyer Realty Holdings LLC (“Sawyer”) in the TCT acquisition”, which understandably appears positive to a classifier. This is because the mere occurrence of the words “acquisition” and “announced” in any order is sufficient to trigger the trained model, which uses only binary bag-of-words (order insensitive) representation. The VOTING strategy thus combines the best of breed results from CONTENT, TITLE, and 1SEN methods.

Apart from deciding the best classification strategy, another significance of this experiment is that it increased the precision of Google’s returned news alerts from 51.4% to 85%, a more than 30% improvement. All in all, the high precision and recall figures confirmed that the *Google Acquisition* dataset is ideally suitable for modeling the $t_{2,3}$ for subsequent AED evaluations in Section 5.6.3.

Online Detection of Anticipatory Event Transition on the *Acquisition7* dataset

In this section, we test the generic classification model trained by *Google Acquisition* for AED on the *Acquisition7* dataset.

Figure 5.22 shows the $t_{2,3}$ boundary for each of the 7 topics of *Acquisition7* distributed along a timeline.

In Figures 5.23-5.29, results of detection of AET are shown for each acquisition topic. Only the first story of e_3 needs to be detected, with all subsequent news articles automatically labeled post- $t_{2,3}$ by the system.

Table 5.15 gives a summary of the overall performances shown in Figures 5.23-5.29. To summarize, AED based on the VOTING method generated 4 accurate alarms, 1 delayed alarm, 2 false alarms and 0 misses. This means that the model trained by *Google Acquisition* was able to uncover the main characteristics of all 7 acquisition topics.

Moreover, comparing this results with that of the CONTENT method as shown in Table 5.16, we found that the AED evaluation for *Acquisition7* dataset is inconsistent with the two-fold cross-validation results for *Google Acquisition* dataset with respect to false alarms. In *Google Acquisition*, the VOTING method reduces false alarms, with the CONTENT method yielding the highest recall. The situation is completely reversed for *Acquisition7*. Based on the analysis in Section 5.6.3, we are inclined to trust the evaluation

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

Figure 5.22: $t_{2,3}$ boundaries in *Acquisition7* dataset.

results for *Google Acquisition* better because the inconsistencies could simply be caused by the relatively small size of the *Acquisition7* dataset.

Nevertheless, in spite of the above inconsistencies, AED results achieved by both the VOTING and CONTENT methods were leaps and bounds ahead of the cosine similarity results (Figures 5.20) and event-conditioned novelty detection results (Table 5.13). This is actually a very encouraging outcome for a pioneering investigation into AED, and thus

Figure 5.23: Online AED evaluation on acquisition event: Adobe acquire Macromedia. $t(d_m) = t(e_3)$, an accurate alarm.

Figure 5.24: Online AED evaluation on acquisition event: CNPC acquires PKZ. $t(d_m) = t(e_3)$, an accurate alarm.

provides strong support and credibility to our AED model and solution.

5.7 Summary

In this chapter, we focus on analyzing the intra-structure of a given topic genre by tracking a user specified event transition, coining the name Anticipatory Event Detection (AED). AED

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

Figure 5.25: Online AED evaluation on acquisition event: eBay acquires Skype. $t(d_m) = t(e_3)$, an accurate alarm.

Figure 5.26: Online AED evaluation on acquisition event: Lenovo acquires IBM PC Division. $t(d_m) < t(e_3)$, a false alarm.

is a more refined and personalized form of event tracking and detection. We investigated suitable document representations and various classification methods to tackle the AED problem, with a substantial focus on improving the AED transition models, which were verified experimentally on two specialized domains.

AED has an essential application in news alerts system, which can help information analysts monitor only relevant events. The holy grail of AED is to detect any number

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

Figure 5.27: Online AED evaluation on acquisition event: Oracle acquires PeopleSoft. $t(d_m) < t(e_3)$, a false alarm.

Figure 5.28: Online AED evaluation on acquisition event: Oracle acquires Siebel. $t(d_m) = t(e_3)$, an accurate alarm.

of AE transitions of arbitrary genres. This is akin to having a live assistant constantly scanning newsfeed monitoring a set of AEs. Our current contributions have achieved the goal of verifying the feasibility of training AE transitional models for homogenous future events.

The main limitation of AED lies in its reliance on a pre-trained transition model for every user-specified anticipatory event. This means that in practice, a user is not allowed to

CHAPTER 5. INTRA-TOPIC TRANSITION DETECTION VIA CLASSIFICATION

Figure 5.29: Online AED evaluation on acquisition event: SBC acquires AT&T. $t(d_m) > t(e_3)$, a delayed alarm.

Table 5.15: Summary of AED results on detecting $t_{2,3}$ for *Acquisition7* using the VOTING method.

<i>Alarms</i>	<i>Accurate</i>	<i>Delayed</i>	<i>False</i>	<i>Miss</i>
Adobe acquires Macromedia	✓			
CNPC acquires PetroKazakhstan (PKZ)	✓			
eBay acquires Skype	✓			
Lenovo acquires IBM PC Division				✓
Oracle acquires PeopleSoft				✓
Oracle acquires Siebel	✓			
SBC acquires AT&T		✓		

specify any anticipatory event, but instead must choose from a list of available pre-trained anticipatory event transitions and ETGs, e.g., terrorist bombing, earthquake disaster, mergers and acquisitions, sports scores, etc. The flip-side of this is that accurate ETG can be built for topics that matters the most to information analysts, and extremely accurate detection rates can be achieved.

Table 5.16: Summary of AED results on detecting $t_{2,3}$ for *Acquisition7* using the CONTENT method.

<i>Alarms</i>	<i>Accurate</i>	<i>Delayed</i>	<i>False</i>	<i>Miss</i>
Adobe acquires Macromedia	✓			
CNPC acquires PetroKazakhstan (PKZ)	✓			
eBay acquires Skype	✓			
Lenovo acquires IBM PC Division				✓
Oracle acquires PeopleSoft		✓		
Oracle acquires Siebel	✓			
SBC acquires AT&T		✓		

Chapter 6

Conclusion

6.1 Discussion

This dissertation studies the topic detection problem for temporal unstructured text streams like news. Currently available methods for detecting topics from news stream are far from satisfactory, partially because the vast majority of existing work attack the problem from the perspective of a traditional clustering framework, overlooking the time dimension, which plays a critical role in streaming data. For example, researchers exhaustively tested many variations of Single-Link and Single-Pass clustering for online topic detection and empirically decided on some reasonable document representation schemes. The rich family of k -component clustering (either flat or hierarchical) was also tailored towards offline topic detection. We categorize all of the above existing work as non-probabilistic models in this dissertation. In fact, there are problems associated even with the most sophisticated non-probabilistic model, namely those of difficult parameter selection, high false alarm rates, low efficiency, and inadequate topic representation. These factors led to my initial idea of conducting feature selection for latent topics, since the very high dimensional Euclidean space would largely add to the difficulty of developing an efficient algorithm.

Unfortunately, we cannot coin a simple feature selection model for topic detection as the problem is typically unsupervised and time-dependant. Ideally, we expect that there exists a feature set in the text collection, with which all documents can be topically discriminated from each other with minimum error. However, hitherto there has been no theoretically consistent and elegant method for identifying such features; the available list of discriminative models, which includes Linear Discriminative Analysis, Support Vector Machine, Boosting, and Conditional Random Fields, etc., all require training data and are thus not suitable for large-scaled online topic detection.

CHAPTER 6. CONCLUSION

With this in mind, we developed a systematic method to categorize all word features into 5 general types using techniques from signal processing in Chapter 3, which serves as one of the major highlights and contributions of this dissertation. The key idea of this work lies in the observations that (a)periodic topics have (a)periodic discriminative features and (un)important topics have (in)active discriminative features, differentiated by their power spectrums and time periods. We were the first to discover many novel interesting characteristics about news streams that were previously unknown, e.g., the different distributions of stopwords during weekdays and weekends, the burstiness of discriminative features and the periodicity of topics, etc. Armed with a small set of extracted discriminative bursty features from offline or online news streams, we proposed a number of effective algorithms to detect topics from a news stream in both offline and online modes, where each topic is depicted by a set of discriminative bursty features.

The solid theoretical foundation of discriminative features on document-level topic detection models was established in Chapter 4. We show the equivalence of a naive Bayesian-based probabilistic model (DIS) to the clustering process of a variation of TFIDF under the condition that only discriminative word features are used. Accordingly, we thoroughly investigate the DIS model on both offline and online topic detection problems for news stream, and benchmarked DIS against the classical non-probabilistic models, and the more complicated generative models like vMF mixture and LDA, etc. The burstiness heuristic adopted in Chapter 3 for identifying the discriminative features was successfully applied here. Our results led to a well-known principle in the area of model selection, which is named Occam’s Razor: sophisticated models like vMF or LDA may shine in more complicated tasks or problems, but for the straightforward problem of topic detection, a simple and mathematically elegant model like DIS can be surprisingly effective and practical.

Lastly, we realized that in practice topic detection is far more complicated than a simple problem of clustering documents into independent (or equally important) topics. A typical Internet user may only subscribe to some specific anticipatory event embedded in a topic from a news alert system. Such an application treats each and every event within a topic genre to be strongly time-dependant. However, analyzing the event transitions within a topic genre is extremely hard, domain-based, and has never been tackled before. We defined this problem as Anticipatory Event Detection (AED) in the dissertation. A case study of a personalized news alert application is described in Chapter 5, where we show how a simple

supervised event transition classifier can be used to effectively identify anticipatory events like mergers and acquisitions. Our preliminary work verifies the importance of mining domain knowledge (i.e., discriminated by named entity types) on AED.

By bridging the gap between theoretical and practical aspects of topic detection, this dissertation opens up several new research frontiers including signal processing of word feature trajectories, utilizing temporal bursty and discriminative features, online detection of burst using a combination of both time and frequency domain analysis, the foundation of which is the systematic and theoretical inclusion of time.

We envisage our work to serve as a useful reference for other researchers in the TDT field. We hope our work can inspire other researchers to try yet more interesting techniques from other fields to further improve the performance of TDT in the many years to come. Who knows, Yahoo may one day do away with human operators, and rely completely on automated TDT algorithms to accurately perform whatever tasks that is requested by users. And we believe we are a step closer to this holy grail, as a result of our research.

6.2 Future Directions

There is scope for much further research in the area of topic detection. Some possible future research directions are listed in the following subsections, grouped by issue type.

6.2.1 Discriminative Word Feature Selection

- The basic Fourier transform is unable to combine time and frequency, as additional mixture density models were used to detect the time location of the dominant burst for word feature trajectories. In the future, time-aware techniques like the wavelet transform can be used.
- Currently we only allowed the one-way state transition from *aperiodic stable* to *periodic stable* in the 3-state automaton model, which limits our detection to newly created periodic bursts. In practice, some periodic burst may die down, e.g. cancelling of weekly soccer matches. In such cases, a fall back mechanism can be put in place to transition from a periodic burst to an aperiodic burst, which eventually dies.

6.2.2 Probabilistic Document-Level Models

- Most existing probabilistic topic detection models assume the dataset to be a mixture of equally important topics. We can instead develop hierarchical probabilistic models, where given a document we can compute the tree path probability instead of the flat likelihood of a leaf node.
- Whenever a news topic is released, in addition to the news content, we could also detect opinions/sentiments of various agencies/countries/persons on the same topic. We can accordingly build opinion probabilistic models for a specific topic type. This work is feasible since discriminative word features (i.e., n-gram may be required rather than the bag-of-words) for opinions and sentiments are limited and can be pre-trained.

6.2.3 Anticipatory Event Detection

- Ensemble learning of discriminative features for user anticipatory event transition is preferred to the exclusive TFIDF-based training. There are many classification features that we could explore further, e.g., the article structure, time, location, and word transition states, etc.
- The current event transition learning heavily relies on the pre-constructed event transition graph. Building such a graph in an unsupervised manner is required in practice. Practical topics from the same topic genre can also exhibit different event transitions. More generally, we can build a Hidden Markov Model or Conditional Random Field model for some trained topics, and apply it to other testing topic within/outside the trained topic types. We may need to consider topic type adaption during model learning.
- User-oriented news alert is extremely useful for Internet marketing including online advertising etc. Tracking what a user subscribes and clicks can result in a more personalized news alert system. We can thus develop smart news aggregation systems that automatically learns a user's topic preferences.

Appendix A

List of Publications

During the three and a half year of my Ph.D. candidature, I have published in top international conferences including SIGIR, SIGKDD (best application paper award), ICDE, ER, WWW, ICDM, and SDM. I was also one of two recipients in Singapore to receive a Microsoft Research Asia Fellowship in 2006 for work done in this thesis. Publications directly related to this thesis are listed as follows.

Journal Publications Under Review

- **Qi He**, Kuiyu Chang, Ee-Peng Lim and Arindam Banerjee. Keep It Simple with Time: A re-examination of Probabilistic Topic Detection Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, submitted.
- **Qi He**, Kuiyu Chang and Ee-Peng Lim. Analyzing Feature Trajectories for Batch and Online Event Detection, *ACM Transactions on Information Systems (TOIS)*, submitted.

Journal and Book Chapter Publications

- **Qi He**, Kuiyu Chang and Ee-Peng Lim. Anticipatory Event Detection via Classification, *Journal of Information Systems and e-Business*, Springer Verlag, vol. 5, no. 3, pp. 275-294.
- **Qi He**, Kuiyu Chang and Ee-Peng Lim. Modeling Anticipatory Event Transitions, *Intelligent and Security Informatics: Techniques and Applications*, Studies in Computation Intelligence, Vol. 135, H. Chen, C. Yang (Editors), Springer Verlag, pp. 97-119

CHAPTER A. LIST OF PUBLICATIONS

Conference Publications

- **Qi He**, Kuiyu Chang and Ee-Peng Lim. Analyzing Feature Trajectories for Event Detection, *the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.
- **Qi He**, Kuiyu Chang, Ee-Peng Lim and Jun Zhang. Bursty Feature Representation for Clustering Text Streams, *the SIAM International Conference on Data Mining (SDM)*, 2007.
- **Qi He**, Kuiyu Chang and Ee-Peng Lim. Using Burstiness to Improve Clustering of Topics in News Streams, *the IEEE International Conference on Data Mining (ICDM)*, 2007.
- R. Aminuddin, R. Suri, Kuiyu Chang, Z. Zainudin, **Qi He** and Ee-Peng Lim. Modeling Anticipatory Event Transitions, *the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2007.
- Kuiyu Chang, R. Aminuddin, R. Suri, **Qi He** and Ee-Peng Lim. Anticipatory Event Detection for Bursty Events, *the Pacific Asia Workshop on Intelligence and Security Informatics (PAISI)*, 2007.
- **Qi He**, Kuiyu Chang and Ee-Peng Lim. A Model for Anticipatory Event Detection, *the 25th International Conference on Conceptual Modeling (ER)*, 2006.
- **Qi He**, Kuiyu Chang and Ee-Peng Lim. Anticipatory Event Detection via Sentence Classification, *the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2006.
- K. Chua, W. S. Ong, **Q. He**, K. Chang and A. Kek. Intelligent Portal for Event-triggered SMS Alerts, *the 2nd IEE Mobility Conference*, 2005.

References

- [ACD⁺98] James Allan, Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the Broadcast News Understanding and Transcription Workshop*, pages 194–218, 1998.
- [AE99] Kjersti Aas and Line Eikvil. Text categorisation: A survey. *Technical report, Norwegian Computing Center*, 1999.
- [AGK01a] James Allan, Rahul Gupta, and Vikas Khandelal. Temporal summaries of news topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–18, 2001.
- [AGK01b] James Allan, Rahul Gupta, and Vikas Khandelal. Topic models for summarizing novelty. In *Proceedings of Workshop on Language Modeling in Information Retrieval*, pages 66–71, 2001.
- [AHF⁺05] James Allan, Stephen Harding, David Fisher, Alvaro Bolivar, Sergio Guzman-Lara, and Peter Amstutz. Taking topic detection from evaluation to practice. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), Track 4*, volume 04, 2005.
- [AHWY03] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 10–18, 2003.
- [AJR⁺99] James Allan, Hubert Jin, Martin Rajman, Charles Wayne, Daniel Gildea, Victor Lavrenko, Rose Hoberman, and David Caputo. Topic-based novelty detection: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1999.

REFERENCES

- [ALJ00] James Allan, Victor Lavrenko, and Hubert Jin. First story detection in tdt is hard. In *Proceedings of the 9th ACM conference on Information and knowledge management (CIKM)*, pages 374–381, 2000.
- [All02a] James Allan. Detection as multi-topic tracking. *Information Retrieval*, 5, 2002.
- [All02b] James Allan. Topic detection and tracking. event-based information organization. *Kluwer Academic Publishers*, 2002.
- [ALMS00] James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. Detections, bounds, and timelines: Umass and tdt-3. In *Proceedings of Topic Detection and Tracking (TDT-3)*, pages 167–174, 2000.
- [APL98] James Allan, Ron Papka, and Victor 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, 1998.
- [Ari76] Roger Ariew. *Ockham’s Razor: A Historical and Philosophical Analysis of Ockham’s Principle of Parsimony*. Champaign-Urbana, University of Illinois, 1976.
- [AWB03] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 314–321, July 28 - August 1 2003.
- [BB07] Arindam Banerjee and Sugato Basu. Topic models over text streams: A study of batch and online unsupervised learning. In *SIAM on data mining*, 2007.
- [BBL99] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine Learning*, 34, 1999.
- [BCF03] Throsten Brants, Francine Chen, and Ayman Farahat. A system for new event detection. In *Proceedings of the 26th annual international ACM SIGIR*

REFERENCES

- conference on Research and development in information retrieval*, pages 330–337, July 28 - August 1 2003.
- [BDGS03] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, and Suvrit Sra. Generative model-based clustering of directional data. In *SIGKDD*, pages 253–264, 2003.
- [BGMV04] Jason Bengel, Susan Gauch, Eera Mittur, and Rajan Vijayaraghavan. Chat-track: Chat room topic detection using classification. In *Proceedings of the 2nd Symposium on Intelligence and Security Informatics*, 2004.
- [BL06] David M. Blei and John D. Lafferty. Dynamic topic models. In *ICML*, 2006.
- [BNJ03] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.
- [BPYC99] Ralf D. Brown, Thomas Pierce, Yiming Yang, and Jaime G. Carbonell. Link detection results and analysis. In *1999 Topic Detection and Tracking (TDT-3) Evaluation Project Workshop*, 1999.
- [BR82] Albert Benveniste and Gabriel Ruget. A measure of the tracking capability of recursive stochastic algorithms with constant gains. *IEEE Transactions on Automatic Control*, 27(3):639–649, 1982.
- [BSW99] Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what’s in a name. *Journal of Machine Learning*, 34(1-3):211–231, 1999.
- [Car] Joseph Carthy. Lexical chains for topic tracking. phd thesis, department of computer science, national university of dublin, 2002.
- [CFB03] Francine Chen, Ayman Farahat, and Thorsten Brants. Story link detection and new event detection are asymmetric. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003*, 2003.

REFERENCES

- [Cie00] Christopher Cieri. Multiple annotations of reusable data resources: Corpora for topic detection and tracking. In *Proceedings Journées Internationales d'Analyse Statistique des Données Textuelles (JADT)*, 2000.
- [CS00] Joe Carthy and Alan Smeaton. The design of a topic tracking system. In *Proceedings of 22nd Annual Colloquim on Information Retrieval Reseach*, pages 10–18, 2000.
- [Das99] S. Dasgupta. Learning mixtures of gaussians. In *IEEE Symposium on Foundations of Computer Science*, 1999.
- [DFM⁺99a] S. Dharanipragada, M. Franz, J.S. McCarley, S. Roukos, and T. Ward. Story segmentation and topic detection for recognized speech. In *Proceedings of 6th European Conference on Speech Communication and Technology (Eurospeech)*, 1999.
- [DFM⁺99b] S. Dharanipragada, M. Franz, J.S. McCarley, S. Roukos, and T. Ward. Story segmentation and topic detection in the broadcast news domain. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.
- [DJ88] R. C. Dubes and A. K. Jain. *Algorithms for Clustering Data*. Prentice Hall, 1988.
- [DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [DM01] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Journal of Machine Learning*, 42(1):143–175, 2001.
- [Elk06] C. Elkan. Clustering documents with an exponentialfamily approximation of the dirichlet compund multinomial distribution. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006.
- [Fis36] R.A. Fisher. The use of multiple measurements in taxonomic problems. In *Annals of Eugenics*, volume 7, pages 179–188, 1936.

REFERENCES

- [FS00] Fumiyo Fukumoto and Yoshimi Suzuki. Event tracking based on domain dependency. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 57–64, 2000.
- [FWMZ01] Martin Franz, Todd Ward, J. Scott McCarley, and Wei-Jing Zhu. Unsupervised and supervised clustering for topic tracking. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 310–317, 2001.
- [FYYL05] G. P. C. Fung, Jeffrey X. Yu, Philip S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *VLDB*, pages 181–192, 2005.
- [GDH04] Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In *Proceedings of the 13rd International World Wide Web Conference (WWW2004)*, 2004.
- [GFS⁺98] Petra Geutner, Michael Finke, Peter Scheytt, Alex Waibel, and Howard Wactlar. Transcribing multilingual broadcast news using hypothesis driven lexical adaptation. In *Proceedings of DARPA Broadcast News Workshop*, 1998.
- [Gho03] J. Ghosh. *Scalable clustering methods for data mining*, In N. Ye (Ed.), *Handbook of data mining*. Lawrence Erlbaum, 2003.
- [GMM⁺03] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O’Callaghan. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 15:515–528, 2003.
- [Gna] Google news alerts, <http://www.google.com/alerts>.
- [Has84] R Hasan. Coherence and cohesive harmonyn. In *J.Flood (ed), Understanding Reading Comprehension, IRA: Newark, Delaware*, 1984.
- [HCL06a] Qi He, Kuiyu Chang, and Ee-Peng Lim. Anticipatory event detection via sentence classification. In *the 2006 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2006.

REFERENCES

- [HCL06b] Qi He, Kuiyu Chang, and Ee-Peng Lim. A model for anticipatory event detection. In *ER*, pages 168–181, 2006.
- [HCL07] Qi He, Kuiyu Chang, and Ee-Peng Lim. Analyzing feature trajectories for event detection. In *the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 207–214, 2007.
- [HCLZ07] Qi He, Kuiyu Chang, Ee-Peng Lim, and Jun Zhang. Bursty feature representation for clustering text streams. In *the SIAM International Conference on Data Mining (SDM)*, 2007.
- [HGM00] Vasileios Hatzivassiloglou, Luis Gravano, and Ankinedu Maganti. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 224–231, 2000.
- [HH76] M Halliday and R Hasan. *Cohesion in English*. Longman, 1976.
- [Hof99] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999.
- [HSC00] Paula Hatch, Nicola Stokes, and Joe Carthy. Topic detection, a new application for lexical chaining? In *Proceedings of BCS-IRSG 2000, the 22nd Annual Colloquium on Information Retrieval Research*, pages 94–103, 2000.
- [I.86] Jolliffe I. *Principal Component Analysis*. NewYork: Spring-Verlag, 1986.
- [JD88] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall advanced reference series. Prentice-Hall, Inc., 1988.
- [JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [Joa97] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pages 143–151, 1997.

REFERENCES

- [JSSW99] Hubert Jin, Rich Schwartz, Sreenivasa Sista, and Frederick Walls. Topic tracking for radio, tv broadcast, and newswire. In *Proceedings of the DARPA Broadcast News Workshop*, pages 199–204, 1999.
- [KA04] Giridhar Kumaran and James 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, 2004.
- [KBDG04] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the 30th annual international VLDB conference*, 2004.
- [Kle02] Jon Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 91–101, 2002.
- [Kli04] Ralf Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Special Issue on Incremental Learning Systems Capable of Dealing with Concept Drift*, 8:281–300, 2004.
- [KNRT05] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. *World Wide Web: Internet and Web Information Systems*, 8:159–178, 2005.
- [KPR90] Anthony Kuh, Thomas Petsche, and Ronald L. Rivest. Learning time-varying concepts. In *Proceedings of the 1990 conference on Advances in neural information processing systems (NIPS) 3*, pages 183–189, 1990.
- [KR90] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.
- [Kwo98] James Tin-Yau Kwok. Automated text categorization using support vector machine. In *Proceedings of the 5th International Conference on Neural Information Processing (ICONIP)*, pages 347–351, 1998.

REFERENCES

- [Lam78] Simon S. Lam. A new measure for characterizing data traffic. *IEEE Transactions on Communications*, 26:137–140, 1978.
- [Ldc] Ldc: <http://www ldc.upenn.edu/>.
- [luc] Apache lucene-core 2.0.0, <http://lucene.apache.org>.
- [LWLM05] Zhiwei Li, Bin Wang, Mingjing Li, and Wei-Ying Ma. A probabilistic model for retrospective news event detection. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 106–113, 2005.
- [Mak] J. Makkonen. Investigations on event evolution in tdt. In *HLT-NAACL 2003*, pages 43–48.
- [Mal99] Stephane Mallat. *A Wavelet Tour of Signal Processing (2nd edition)*. Academic Press, 1999.
- [MBJ99] Katharina Morik, Peter Brockhausen, and Thorsten Joachimss. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, pages 268–277, 1999.
- [McL04] G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition (New Edition)*. Wiley-Interscience, 2004.
- [MDK⁺97] Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, and Mark Przybocki. The det curve in assessment of detection task performance. In *Proceedings of Eurospeech*, pages 1895–1898, 1997.
- [MI90] George Miller and Special Issue. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4), 1990.
- [Mla98] D. Mladenic. Feature subset selection in text-learning. In *Proceedings of ECML'98*, pages 95–100. Springer-Verlag, 1998.

REFERENCES

- [MLRM04] Ana Maguitman, David Leake, Thomas Reichherzer, and Filippo Menczer. Dynamic extraction topic descriptors and discriminators: towards automatic context-based topic search. In *Proceedings of the 13rd ACM international conference on Information and knowledge management (CIKM)*, pages 463–472, 2004.
- [MN98] A. K. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceeding of Workshop on Text Categorization (AAAI’98)*, pages 41–48, July 1998.
- [MRW⁺99] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Muller. Fisher’s discriminant analysis with kernels. In *Proceedings of IEEE Neural Networks for Signal Processing Workshop*, 1999.
- [MZ05] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 198–207, 2005.
- [New87] Isaac Newton. *Philosophiae Naturalis Principia Mathematica*. 1687.
- [NFPA04] Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. Event threading within news topics. In *Proceedings of the 13rd ACM international conference on Information and knowledge management (CIKM)*, pages 446–453, 2004.
- [NMTM00] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39:103–134, 2000.
- [oaD05] Włodzisław Duch. Computational intelligence: Methods and applications, lecture 7: Discriminant component analysis, 2005.
- [Oar99] Douglas W. Oard. Tracking with the prize information retrieval system. In *Proceedings of the DARPA Broadcast News Workshop*, pages 209–211, 1999.
- [Pen01] William D. Penny. Kullback-liebler divergences of normal, gamma, dirichlet and wishart densities. *Technical report*, 2001.

REFERENCES

- [RA05] Hema Raghavan and James Allan. Passage feedback for news tracking. In *CIIR Technical Report*, 2005.
- [RWHB⁺96] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aaron Gull, and Marianna Lau. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30, 1996.
- [S.A99] S.A.Lowe. The beta-binomial mixture model and its application to tdt tracking and detection. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.
- [SA00] Russell Swan and James Allan. Automatic generation of overview timelines. In *Proceedings of the Special Interest Group in Information Retrieval*, pages 47–55, 2000.
- [SB88] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.
- [SC01] Nicola Stokes and Joe Carthy. Combining semantic and syntactic document classifiers to improve first story detection. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 424–425, 2001.
- [SG02] A. Strehl and J. Ghosh. Cluster ensemblesa knowledge reuse framework for combining partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.
- [SL99] J. Michael Schultz and Mark Liberman. Topic detection and tracking using idf-weighted cosine coefficient. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.
- [Spi82] Jeffrey R. Spirn. Network modeling with bursty traffic and finite buffer space. In *Proceedings of the Computer Network Performance Symposium*, pages 21–28, 1982.
- [Spr] Statistical pattern recognition toolbox: <http://cmp.felk.cvut.cz/~xfrancv/stprtool/>.
- [SRHG86] Jeffrey C. Schlimmer and Jr. Richard H. Granger. Incremental learning from noisy data. *Machine Learning*, 1:317–354, 1986.

REFERENCES

- [Svm] Svm-light, <http://svmlight.joachims.org/>.
- [Tdt] Tdt: Annotation manual version 1.2, august 4 2004, <http://www ldc.upenn.edu/projects/tdt2004>.
- [TSHTS01] Gokhan Tur, Andreas Stolcke, Dilek Hakkani-Tur, and Elizabeth Shriberg. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, 27, 2001.
- [Tsy04] Alexey Tsymbal. The problem of concept drift: Definitions and related work. *Technical report, Department of Computer Science, Trinity College*, 2004.
- [VMVG04] Michail Vlachos, Christopher Meek, Zografoula Vagena, and Dimitrios Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *SIGMOD*, pages 131–142, 2004.
- [Wik] Wikipedia: http://en.wikipedia.org/wiki/naive_bayes_classifier.
- [WJSS99] Frederick Walls, Hubert Jin, Sreenivasa Sista, and Richard Schwartz. Topic detection in broadcast news. In *Proceedings of the DARPA Broadcast News Workshop*, pages 193–198, 1999.
- [YAPL00] Yiming Yang, Thomas Ault, Thomas Pierce, and Charles W. Lattimer. Improving text categorization methods for event tracking. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 65–72, 2000.
- [YCG⁺99] J. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. Topic tracking in a news stream. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.
- [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, 1998.
- [YS06] Christopher C. Yang and Xiaodong Shi. Discovering event evolution graphs from newswires. In *WWW*, pages 945–946, 2006.

REFERENCES

- [YZCJ02] Yiming Yang, Jian Zhang, Jaime Carbonell, and Chun Jin. Topic-conditioned novelty detection. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 688–693, 2002.
- [ZGY05] Jian Zhang, Zoubin Ghahramani, and Yiming Yang. A probabilistic model for online document clustering with application to novelty detection. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1617–1624, 2005.
- [ZLBM06] Qiankun Zhao, Tie-Yan Liu, Sourav S. Bhowmick, and Wei-Ying Ma. Event detection from evolution of click-through data. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 484–493, 2006.
- [ZS03] Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 336–345, 2003.