

Fachbereich 4: Informatik



# Mining Social Media: Methods and Approaches for Content Analysis

Vom Promotionsausschuss des Fachbereichs 4: Informatik  
an der Universität Koblenz-Landau  
zur Verleihung des akademischen Grades  
Doktor der Naturwissenschaften (Dr. rer. nat.)  
genehmigte

DISSERTATION

von

Nasir Naveed  
M.S. in Computer Science

Koblenz - 2013

Datum der Promotion: 03.07.2013

Vorsitz des Promotionsausschusses: Prof. Dr. R. Grimm  
Vorsitz der Promotionskommission: Prof. Dr. P. Schubert  
1. Berichterstatter: Prof. Dr. S. Staab  
2. Berichterstatter: Dr. Y. He

Veröffentlicht als Dissertation der Universität Koblenz-Landau.



# Erklärung

Hiermit erkläre ich gemäß §8 der Promotionsordnung des Fachbereichs 4: Informatik der Universität Koblenz-Landau,

- dass ich die vorliegende Dissertation mit dem Titel “*Mining Social Media: Methods and Approaches for Content Analysis*” selbst angefertigt und alle benutzten Hilfsmittel in der Arbeit angegeben habe,
- dass ich die Dissertation oder Teile der Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe, und
- dass ich weder diese noch eine andere Abhandlung bei einer anderen Hochschule als Dissertation eingereicht habe.

Koblenz, den 22 April, 2013

Nasir Naveed



# Acknowledgment

First of all I thank the Almighty Allah for giving me the strength and health to conduct and successfully complete my PhD. Then I would like to thank my parents who spent a large part of their lives and resources for grooming and educating me and to my family members namely Bashir Ramay, Jamil, Attiya, Manahil and Ahmad for providing moral and emotional support at all times.

I would like to acknowledge my advisor Prof. Dr. Steffen Staab for guiding, motivating and providing every kind of support throughout the course of my studies. I have always found him very kind, helpful and encouraging. I would also like to acknowledge Dr. Yulan He for reviewing my thesis and providing useful feedback. My PhD would not have been possible without the financial support of Higher Education Commission (HEC) of Pakistan and the organizational support of Deutsche Akademische Austausch Dienst (DAAD). The last year of my PhD was funded by the EU project WeGov, I am really thankful to them.

All of my colleagues especially Dr. Thomas Gottron at the WeST institute made the PhD experience a lot wonderful. Their feedback has always helped in improving my skills and research. I really enjoyed every moment that I had spent with them. I would also like to acknowledge Dr. Sergej Sizov, Dr. Jérôme Kunegis and Arifah Che Alhadi who partially contributed to this research work.

My friends Dr. Rabeeh Abbasi, Ingrid Kantorova, Dr. Masroor Elahi Babar and Tariq Parvez, were like a family to me. They were with me and my family at all the occasions of happiness and sorrow. Throughout the duration of PhD, the staff members Sabine Hülstrunk, Silke Werger, Ruth Ehrenstein, and Ute Lenz-Perscheid had always been there to help me.



# Abstract

Web 2.0 provides technologies for online collaboration of users as well as the creation, publication and sharing of user-generated contents in an interactive way. Twitter, CNET, CiteSeerX, etc. are examples of Web 2.0 platforms which facilitate users in these activities and are viewed as rich sources of information. In the platforms mentioned as examples, users can participate in discussions, comment others, provide feedback on various issues, publish articles and write blogs, thereby producing a high volume of unstructured data which at the same time leads to an information overload. To satisfy various types of human information needs arising from the purpose and nature of the platforms requires methods for appropriate aggregation and automatic analysis of this unstructured data. In this thesis, we propose methods which attempt to overcome the problem of information overload and help in satisfying user information needs in three scenarios.

To this end, first we look at two of the main challenges of sparsity and content quality in Twitter and how these challenges can influence standard retrieval models. We analyze and identify Twitter content features that reflect high quality information. Based on this analysis we introduce the concept of “interestingness” as a static quality measure. We empirically show that our proposed measure helps in retrieving and filtering high quality information in Twitter. Our second contribution relates to the content diversification problem in a collaborative social environment, where the motive of the end user is to gain a comprehensive overview of the pros and cons of a discussion track which results from social collaboration of the people. For this purpose, we develop the FREuD approach which aims at solving the content diversification problem by combining latent semantic analysis with sentiment estimation approaches. Our evaluation results show that the FREuD approach provides a representative overview of sub-topics and aspects of discussions, characteristic user sentiments under different aspects, and reasons expressed by different opponents. Our third contribution presents a novel probabilistic Author-Topic-Time model, which aims at mining topical trends and user interests from social media. Our approach solves this problem by means of Bayesian modeling of relations between authors, latent topics and temporal information. We present results of application of the model to the scientific publication datasets from CiteSeerX showing im-

proved semantically cohesive topic detection and capturing shifts in authors' interest in relation to topic evolution.



# Zusammenfassung

Das Web 2.0 stellt online Technologien zur Verfügung, die es Nutzern erlaubt gemeinsam Inhalte zu erstellen, zu publizieren und zu teilen. Dienste wie Twitter, CNet, CiteSeerX etc. sind Beispiele für Web 2.0 Plattformen, die zum einen Benutzern bei den oben beschriebenen Aktivitäten unterstützen und zum anderen als Quellen reichhaltiger Information angesehen werden können. Diese Plattformen ermöglichen es Nutzern an Diskussionen teilzunehmen, Inhalte anderer Nutzer zu kommentieren, generell Feedback zu geben (z.B. zu einem Produkt) und Inhalte zu publizieren, sei es im Rahmen eines Blogs oder eines wissenschaftlichen Artikels. Alle diese Aktivitäten führen zu einer großen Menge an unstrukturierten Daten. In diesem Überfluss an Informationen kann auf den persönlichen Informationsbedarf einzelner Benutzer nicht mehr individuell genug eingegangen werden kann. Methoden zur automatischen Analyse und Aggregation unstrukturierter Daten die von einzelnen Plattformen zur Verfügung gestellt werden, können dabei helfen den sich aus dem unterschiedlichen Kontext der Plattformen ergebenden Informationsbedarf zu beantworten. In dieser Arbeit stellen wir drei Methoden vor, die helfen den Informationsüberfluss zu verringern und es somit ermöglichen den Informationsbedarf einzelner Nutzer besser zu beantworten.

Der erste Beitrag dieser Arbeit betrachtet die zwei Hauptprobleme des Dienstes Twitter: die Kürze und die Qualität der Einträge und wie sich diese auf die Ergebnisse von Suchverfahren auswirken. Wir analysieren und identifizieren Merkmale für einzelne Kurznachrichten auch Twitter (sog. Tweets), die es ermöglichen die Qualität eines Tweets zu bestimmen. Basierend auf dieser Analyse führen wir den Begriff "Interestingness" ein, der als statisches Qualitätsmaß für Tweets dient. In einer empirischen Analyse zeigen wir, dass die vorgeschlagenen Maße dabei helfen qualitativ hochwertigere Information in Twitter zu finden und zu filtern. Der zweite Beitrag beschäftigt sich mit dem Problem der Inhaltsdiversifikation in einem kollaborativen sozialen System, z.B. einer online Diskussion die aus der sozialen Kollaboration der Nutzer einer Plattform entstanden ist. Ein Leser einer solchen Diskussion möchte sich einen schnellen und umfassenden Überblick über die Pro und Contra Argumente in der Diskussion verschaffen. Zu diesem Zweck wurde FREuD entwickelt, ein Ansatz der hilft das Diversifikationsproblem von In-

halten in den Griff zu bekommen. FREuD kombiniert Latent Semantic Analysis mit Sentiment Analyse. Die Evaluation von FREuD hat gezeigt, dass es mit diesem Ansatz möglich ist, einen umfassenden Überblick über die Unterthemen und die Aspekte einer Diskussion, sowie über die Meinungen der Diskussteilnehmer zu liefern. Der dritte Beitrag dieser Arbeit ist ein neues Autoren-Thema-Zeit Modell, das es ermöglicht Trendthemen und Benutzerinteressen in sozialen Medien zu erfassen. Der Ansatz löst dieses Problem indem er die Relationen zwischen Autoren, latenter Themen und zeitlicher Information mittels Bayes'schen Netzen modelliert. Unsere Evaluation zeigt eine verbesserte Erkennung von semantisch zusammenhängenden Themen und liefert im weiteren Informationen darüber in wie weit die Veränderung im Interesse einzelner Autoren mit der Entwicklung einzelner Themengebiete zusammenhängt.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Scenario 1: Quality Features and Retrieval in Mircoblogs . . .	2
1.1.1	Methodology . . . . .	3
1.1.2	Research Contribution . . . . .	3
1.2	Scenario 2: Social Content Diversification . . . . .	4
1.2.1	Methodology . . . . .	4
1.2.2	Research Contribution . . . . .	5
1.3	Scenario 3: Mining User Interests from Social Contents . . .	5
1.3.1	Methodology . . . . .	6
1.3.2	Research Contribution . . . . .	7
1.4	Thesis Structure . . . . .	7
1.5	Dissemination . . . . .	8
<b>2</b>	<b>Foundations of Information Retrieval</b>	<b>11</b>
2.1	Information Retrieval Models: an Overview . . . . .	11
2.1.1	Formal definition of an IR Model . . . . .	12
2.1.2	Boolean Model . . . . .	13
2.1.3	Vector Space Model . . . . .	14
2.1.4	Extended Boolean Model . . . . .	17
2.1.5	Probabilistic Models . . . . .	17
2.2	Evaluation Measures . . . . .	19
2.2.1	Precision and Recall . . . . .	20
2.2.2	Average Precision . . . . .	20
2.2.3	Mean Average Precision ( <i>MAP</i> ) . . . . .	21
2.2.4	Normalized Discounted Cumulative Gain ( <i>nDCG</i> ) . . .	21
2.3	Content Diversification . . . . .	22
2.3.1	Diversity Evaluation Measures . . . . .	23
2.3.2	$\alpha$ - <i>nDCG</i> . . . . .	23
2.3.3	Intent Aware (IA) Metrics . . . . .	25
2.4	Summary . . . . .	25

<b>3</b>	<b>Foundations of Probabilistic Modeling</b>	<b>27</b>
3.1	Graphical Models . . . . .	28
3.2	Bayesian Networks (Representation) . . . . .	29
3.2.1	Generative Models . . . . .	32
3.3	Parameter Estimation Techniques (Learning) . . . . .	33
3.3.1	Maximum Likelihood Estimation (MLE) . . . . .	35
3.3.2	Maximum a posteriori Estimation (MAP) . . . . .	36
3.3.3	Bayesian Estimation . . . . .	36
3.4	Inference in Probabilistic Models . . . . .	37
3.4.1	Markov Chain Monte Carlo Methods (MCMC) . . . . .	37
3.4.2	Gibbs Sampling . . . . .	39
3.5	Evaluating Topic Models . . . . .	40
3.5.1	Querying . . . . .	40
3.5.2	Perplexity . . . . .	42
3.6	Summary . . . . .	43
<b>4</b>	<b>Searching Microblogs: Coping with Document Quality and Sparsity</b>	<b>45</b>
4.1	The Microblog Environment . . . . .	46
4.2	Retweet Datasets . . . . .	48
4.3	Content-based Retweet Prediction . . . . .	49
4.3.1	Features . . . . .	50
4.3.2	Regression Analysis . . . . .	54
4.3.3	Accuracy of Retweet Prediction . . . . .	54
4.3.4	Analysis of the Weights . . . . .	55
4.3.5	Example: Interesting Tweets . . . . .	59
4.4	Retrieval on Microblogs . . . . .	60
4.4.1	Term Sparsity and Length Normalization . . . . .	61
4.4.2	“Interestingness” as Static Quality Measure . . . . .	63
4.5	Applications and Evaluation . . . . .	64
4.6	Ranking Microblogs . . . . .	64
4.6.1	Retrieval System Setup . . . . .	64
4.6.2	Evaluation Method . . . . .	65
4.6.3	Evaluation Results . . . . .	67
4.7	Filtering Microblogs . . . . .	70
4.7.1	Dataset . . . . .	72
4.7.2	Evaluation Criteria . . . . .	72
4.7.3	LiveTweet System . . . . .	73
4.7.4	Evaluation Results . . . . .	75
4.8	Related Work . . . . .	78
4.9	Summary . . . . .	82

<b>5</b>	<b>Feature Sentiment Diversification of User Generated Contents: The FREuD Approach</b>	<b>85</b>
5.1	Formal Task Definition . . . . .	87
5.2	The FREuD approach . . . . .	88
5.2.1	Feature Extraction . . . . .	89
5.2.2	Sentiment Estimation . . . . .	89
5.2.3	Feature-Sentiment Estimation . . . . .	89
5.2.4	Review Subset Selection . . . . .	90
5.2.5	FREuD Variations . . . . .	91
5.3	Evaluation Setup . . . . .	94
5.3.1	Dataset . . . . .	94
5.3.2	Baseline Systems . . . . .	95
5.3.3	Developing a Gold Standard for the Dataset . . . . .	96
5.3.4	Inter-rater Agreement . . . . .	98
5.3.5	Diversity Evaluation Metrics . . . . .	100
5.4	Experimental Results . . . . .	101
5.5	Related Work . . . . .	106
5.6	Summary . . . . .	108
<b>6</b>	<b>Temporal Dynamics of Topics and Authors in Social Media</b>	<b>109</b>
6.1	The Author-Topic-Time Model (ATT) . . . . .	110
6.1.1	Model Design . . . . .	110
6.1.2	Model Parameters . . . . .	114
6.1.3	Parameters Estimation . . . . .	114
6.2	Experiments and Evaluation . . . . .	117
6.2.1	Dataset . . . . .	117
6.2.2	Evaluation Method . . . . .	117
6.2.3	Evaluation Results . . . . .	118
6.2.4	Application Scenarios . . . . .	123
6.3	Related Work . . . . .	125
6.3.1	Topic Modeling . . . . .	125
6.3.2	Probabilistic Topic Models . . . . .	126
6.3.3	Parameter Estimation . . . . .	129
6.4	Summary . . . . .	129
<b>7</b>	<b>Conclusions</b>	<b>131</b>
7.1	<i>Interestingness</i> : a Measure of Static Content Quality . . . . .	132
7.2	The FREuD Approach . . . . .	133
7.3	Author-Topic-Time Model . . . . .	134
7.4	Outlook . . . . .	134
<b>A</b>	<b>Data Set – the FREuD Approach</b>	<b>137</b>



# List of Figures

2.1	Information retrieval models categorized on their mathematical basis and properties. (Source: [68]) . . . . .	13
2.2	The cosine of $\theta$ is adopted as $sim(d, q)$ . (Source: [127]) . . . . .	16
3.1	A directed graphical model showing the joint probability distribution over three variables $x, y$ , and $z$ reflecting to decomposition in the right hand side of Equation 3.1. (Source: [5], p. 361) . . . . .	29
3.2	Example of a directed acyclic graph describing the joint distribution over variables $t_1, \dots, t_7$ . The corresponding decomposition of the joint distribution is given by Equation 3.3. (Source: [5], p. 362) . . . . .	30
3.3	Bayesian polynomial regression model. . . . .	31
3.4	Directed graphical model representing the joint distribution corresponding (Equation 3.5) to the Bayesian polynomial regression model with the deterministic variables or model parameters are shown as solid circles. (Source: [5], p. 364) . . . . .	32
3.5	A directed graphical model explaining the process by which documents are generated. . . . .	33
3.6	A Language model that specifies how the documents are generated in the real world. . . . .	34
3.7	Bayesian inference reflecting the process for estimating the parameters of the language model specified in Figure3.6. . . . .	34
4.1	Distribution of maximal term frequencies ( $max-tf$ ) in Twitter messages of the CHOUDHURY-EXT dataset after removing stop words. . . . .	47
4.2	The accuracy of retweet prediction using logistic regression based on all features, and of each feature separately, in the PETROVIĆ dataset. The accuracy is represented as a ROC curve. For clarity, ROC curves for topics are not shown. . . . .	56

4.3	The accuracy of retweet prediction using logistic regression based on all features, and of each feature separately, in the CHOUDHURY dataset. The accuracy is represented as a ROC curve. For clarity, ROC curves for topics are not shown. . . . .	56
4.4	The accuracy of retweet prediction using logistic regression based on all features, and of each feature separately, in the CHOUDHURY-EXT dataset. The accuracy is represented as a ROC curve. For clarity, ROC curves for topics are not shown. . . . .	57
4.5	Distribution of document length (in characters) in the CHOUDHURY dataset. . . . .	62
4.6	MAP and P@5 performance: in total and resolved by query length. . . . .	68
4.7	Results for number of times assessors' preferred the resultset from each participating system over the others for different query lengths. . . . .	71
4.8	Evaluation results of LiveTweet at <b>allrel</b> scenario under various performance measures. . . . .	76
4.9	Mean average precision (MAP) scores of LiveTweet at <b>by-rank</b> scenario. . . . .	77
4.10	Mean average precision (MAP) score of LiveTweet for individual topics at <b>by-rank</b> scenario. . . . .	78
4.11	Mean average precision (MAP) score of LiveTweet under individual query groups at <b>allrel</b> scenario. . . . .	78
4.12	Mean average precision (MAP) of LiveTweet against query length for by-rank. . . . .	79
4.13	Mean average precision (MAP) of LiveTweet against query length for by-rank. . . . .	79
4.14	Mean average precision (MAP) of LiveTweet against query length for by-rank. . . . .	80
5.1	Plot showing relationship between review length measured in number of words and associated sentiment score. . . . .	92
5.2	Plot showing relationship between review length and sentiment words used in the review . . . . .	92
5.3	Plot showing relationship between review length measured in number of words to the ratio of positive (p) to negative (n) sentiment scores of the review. . . . .	93
5.4	Plot showing length-wise distribution of reviews in our experimental dataset. . . . .	95
5.5	Plot showing review diversification performance of all approaches aggregated for each product category. . . . .	103
5.6	Plot showing review diversification performance of all approaches for each individual product category. . . . .	105



6.1	Document generation process as specified by the ATT model.	112
6.2	Plot showing average word perplexity scores achieved by ATT and LDA model. Lower perplexity scores indicate better generalization performance. . . . .	119
6.3	Three related Bayesian network models for document generation. . . . .	128
A.1	Screen shot of the FREuDs' evaluation website home page requiring assessor to input a unique id which was used to annotate each review by three unique assessors. . . . .	140
A.2	Screen shot of the FREuDs' evaluation website assessor info page. The assessors were required to select a product from various product categories to annotate the product related reviews. Additionally assessors existing knowledge of the the selected product was also recorded. . . . .	141
A.3	Screen shot of the FREuDs' evaluation website instruction page. This pages provides the details of annotation process and step by step guide for annotating the features and feature related sentiment. . . . .	142
A.4	Screen shot of the FREuDs' evaluation website example task. The example task page provides an example of a completed task according to the instructions given on the instruction page.	143
A.5	Screen shot of the FREuDs' evaluation website actual task page. This page shows the review text from the assessors' selected product and table for recording observations about product feature and sentiment. . . . .	144



# List of Tables

4.1	List of established Twitter datasets used in our experiments.	49
4.2	List of patterns used by the people on Twitter to mark retweets. .....	49
4.3	The features and their value range used to represent tweets.	50
4.4	Terms and emoticons expressing positive and negative emotions in Twitter messages. ....	52
4.5	Weights of features learned by logistic regression on the CHAUDHURY-EXT dataset. Positive values denote a positive contribution to a tweet being retweeted; negative weights denote a negative contribution to a tweet being retweeted. ....	57
4.6	Logistic regression weights and corresponding high probability terms that describe a particular topic in the CHAUDHURY dataset. The weights can be interpreted as the log-odds of a tweet from a given topic to be retweeted. Positive weights indicate topics that are likely to be retweeted and negative weights indicate topics that are unlikely to be retweeted. . . .	59
4.7	Top 10 interesting tweets by the log-odds of predicted retweet probability for the query <code>Recipe</code> in the CHAUDHURY dataset. . .	60
4.8	Queries used to describe microblog information needs. . . . .	66
4.9	Top 10 tweets for the query <code>beer</code> using the <code>Lucene-noLen</code> setting. . . . .	69
4.10	Top 10 tweets for query <code>beer</code> using the <code>Retweet-Odds</code> setting.	70
4.11	Results of number of times assessors preferred resultset from one system over the other systems. . . . .	71
4.12	Length-wise distribution of query topics against their frequency as provide by the TREC 2011 organizers to be used by the participating systems in the retrieval of tweets. . . . .	72
4.13	Statistical test of significance between the performance of <b>WESTfilter</b> and <b>WESTflex</b> at <b>allrel</b> and <b>by-rank</b> scenarios for various measures. . . . .	77

5.1	Topics as determined by LDA from the product reviews approximating the features of Camera, Cellphone and Printer category. . . . .	90
5.2	CNET product review dataset used for evaluating the FREuD approach. . . . .	94
5.3	List of features for Camera, Cellphone and Printer category collected from various websites to be used in the evaluation setup. . . . .	96
5.4	Dataset annotated by the assessors for features and sentiments to be use in the evaluation. . . . .	98
5.5	Statistics of the assessors who participated in the annotation process to obtain gold standard dataset. . . . .	99
5.6	Category-wise inter-rater agreement among assessors over coverage of the features in the reviews. . . . .	100
5.7	Category-wise inter-rater agreement among assessors over feature-sentiment annotations in the reviews. . . . .	100
5.8	Diversification performance comparison of all approaches under individual products in <b>Camera</b> category using $\alpha$ -nDCG@5.101	
5.9	Diversification performance comparison of all approaches under individual products in <b>Cellphone</b> category using $\alpha$ -nDCG@5.102	
5.10	Diversification performance comparison of all approaches under individual products in <b>Printer</b> category using $\alpha$ -nDCG@5.102	
5.11	Relative percentage improvement in $\alpha$ -nDCG scores achieved by FREuD variations against the two baseline systems. . . .	102
5.12	Results showing statistical significance of differences in performance of FREuD and baseline systems using t-test at 5% significance level. . . . .	103
5.13	Diversification performance comparison of all approaches using average $\alpha$ -nDCG@5. under each individual product category. . . . .	104
5.14	Relative percentage improvement in diversification achieved by FREuD variations over the <b>CNET-default</b> baseline. . .	104
5.15	Relative percentage improvement in diversification achieved by FREuD variations over the <b>CNET-diversified</b> baseline. .	104
5.16	Category-wise statistical significance test of performance difference against <i>CNET-default</i> (at 5% significance level). . . .	105
5.17	Category-wise statistical significance test of performance difference against <i>CNET-diversified</i> (at 5% significance level). .	106
6.1	Notation used in the modeling process . . . . .	112
6.2	Representation of 9 topics from a 100-topic run of Gibbs Sampler for CiteSeer dataset discovered by ATT model . . . . .	120
6.3	Representation of 9 topics from CiteSeer dataset discovered by LDA . . . . .	121

6.4	Average KL divergence between topics for ATT and LDA . .	121
6.5	Symmetric KL divergence for pairs of topics shown in Table 6.6, 6.7, 6.8, 6.9 . . . . .	122
6.6	Top terms, influential authors and beta PDF depicting topic life cycle for the topic “Grid Computing“ in CiteSeer dataset	122
6.7	Top terms, influential authors and beta PDF depicting topic life cycle for the topic “Image Analysis“ in CiteSeer dataset .	123
6.8	Top terms, influential authors and beta PDF depicting topic life cycle for the topic “Semantic Web“ in CiteSeer dataset .	124
6.9	Top terms, influential authors and beta PDF depicting topic life cycle for the topic “Database Systems“ in CiteSeer dataset	125
6.10	Intra-topic symmetric KL divergence for different pairs of au- thors . . . . .	125
6.11	Inter-topic symmetric KL divergence for different pairs of au- thors . . . . .	126
A.1	End user product reviews dataset annotated by the assessors for features and associated sentiments. . . . .	137
A.2	Attributes available in metadata collected for each product using CNET developers’ APIs. This information is also pub- lished in the form of xml files in the gold standard dataset for each product. . . . .	138



# Chapter 1

## Introduction

Human generated information which exists in digital books, social media streams, emails messages, pictures, audio, video etc. holds a major share of content produced in today's world. The form in which this information exists makes it unstructured and unlikely to align with rows and columns of a database. This unstructured free form information accounts for 90% of all the information and is growing at a rate of as fast as three times of structured data.

The World Wide Web is being increasingly used as a medium that fosters interaction among people, sharing of experiences and knowledge as well as collaborating group activities. The Web 2.0 provides an interactive platform for content sharing activities where people react to real life events by raising issues, sharing views, participating in discussions, commenting others, and thereby, generate a tremendous amount of online content. Social networking portals such as Twitter<sup>1</sup> and Facebook<sup>2</sup>, online digital libraries such as CiteSeerX<sup>3</sup> and ACM Digital Library, review portals such as CNET<sup>4</sup> and GSM arena are few examples of the platforms that help fostering communication and publishing of information. The pace at which this information is being generated makes it more tedious to access, find and satisfy human information needs. The conclusion drawn by a survey of about 1,000 internet experts conducted and published in July, 2012 [53] states that human and automatic machine analysis of this big data could enhance productivity, improve social, political and economic intelligence. According to the survey results, the analysis of this Big Data can and will help in the development of methods that can find patterns of data to predict outcomes, real-time forecasting of events, and the development of advanced correlations algorithms that enable new and deeper understanding of this data world.

Information generated in each of the example platforms mentioned above

---

<sup>1</sup><http://twitter.com/>

<sup>2</sup><http://www.facebook.com/>

<sup>3</sup><http://citeseerx.ist.psu.edu//>

<sup>4</sup><http://reviews.cnet.com/>

varies for its purpose and intended audience. As reported in [18], Twitter is mainly viewed as an information broadcasting platform used by people for sharing and relaying news about events, issues, and current affairs which are of interest to a general audience, while information generated in review portals is much focused, narrow and reflects the opinion of individual members of the community towards an innovation. Whereas, online digital libraries publish scholarly articles offering information which is usually an output of scientific work and may be of interest to a specific community. Thus, users' information needs in each of the above mentioned platform vary and reflect characteristics of the contents generated in those platforms. Addressing the users' needs in such platforms requires content analysis and retrieval methods optimized to make use of the content features that are typical to each of the platforms.

In this thesis we concentrate on the analysis and categorization of social content features and building algorithms on top that address specific user information needs in example platforms. To this end, we develop methods that improve retrieval quality, diversify contents and capture hidden correlation patterns in the text that can best explain a document collection. We identify and work on three scenarios related to Twitter, CNET and CiteSeerX and empirically show that our proposed methods improve over other classical and state of the art standard content analysis and retrieval methods. The example scenarios are discussed below.

## 1.1 Scenario 1: Quality Features and Retrieval in Mircoblogs

Our first scenario relates to a popular online content sharing service Twitter; a microblogging service having more than 500 million<sup>5</sup> users [30, 115]. Twitter allows users to share information with each other via short messages termed as *Tweets*, producing over 400 million messages a day [115]. In Twitter, users can *follow* other users in order to receive their tweets. If a user considers a tweet interesting, she may forward it to her own followers. This practice is called *retweeting* and usually users retweet the content of general interest or concerned with the audience who follows their tweets [7]. The purpose of retweeting is often to disseminate information to one's followers.

The conciseness of a tweet has been cited as a major reason for the success of Twitter, however at the same time it leads to information overload. The problem of information overload is evident from the high volume of data generated everyday from the wide range of uses of Twitter by its large user base. The quality assessment of the Twitter content is necessary as the microblog documents range from spam over trivia and personal chatter to

---

<sup>5</sup>as of April. 2013



## 1.1. SCENARIO 1: QUALITY FEATURES AND RETRIEVAL IN MICROBLOGS<sup>3</sup>

news broadcasts, self presentation, information dissemination, and reports of current hot topics. Therefore, there is a need to identify microblog document properties that when used help in improving retrieval and filtering of high quality contents in microblogs.

In the context of retrieval on Twitter we address two research questions.

- Our first question is, “Can we use retweet as a function of *interestingness* to develop a model that describes what is of interest on the social network Twitter”?
- The second research question is, “Can we use *interestingness* as a notion of content quality to retrieve high quality contents in Twitter”?

### 1.1.1 Methodology

To tackle the problems of information overload and retrieval of high quality tweets, we think of these as classification tasks based on content features. To this end, we use text pre-processing techniques and probabilistic topic modeling for extracting content features and finding patterns of correlation between document terms. We analyze a set of high- and low-level content based features on a large collection of Twitter messages. The low level features comprise the words contained in a tweet, the tweet being a direct message, the presence of URLs, hashtags, usernames, emoticons, and of question and exclamation marks as well as terms with a strong positive or negative connotation. The high-level features are formed by associating tweets to topics as estimated by Latent Dirichlet Allocation (LDA) [6] and by determining the sentiments of a tweet. We train a prediction model to forecast for a given tweet its likelihood of being retweeted based on its content features. From the parameters learned by the model, we deduce what are the influential content features that contribute to the likelihood of a retweet.

We base our notion of interestingness on the retweet function and use the probability of a tweet being retweeted as an indicator of its static content quality. This notion of a tweet being potentially interesting to other users is thus a suitable way to capture the content quality in Twitter and overcome the problem of retrieving high quality contents.

### 1.1.2 Research Contribution

In answer to the question posed in Section 1.1, we make the following research contributions to the literature.

- We analyze microblogging social media contents such as Twitter and propose a method to identify influential content features that make a tweet interesting and contribute most strongly to the probability of a tweet being retweeted.

- We analyze the problem of content quality in Twitter and introduce *interestingness* as a measure of static content quality. We empirically show that *interestingness* improves retrieving high quality contents in social media.

## 1.2 Scenario 2: Social Content Diversification

Online discussions, user reviews and comments on the Social Web are valuable sources of information about products, services, or shared contents. The rapidly growing popularity and activity of Web communities raises novel questions of appropriate aggregation and diversification of such social contents. Our second scenario relates to aggregation and diversification of social contents to gain a comprehensive overview of various aspects of a discussion in social media platforms. For this purpose, we use reviews from CNET website<sup>6</sup> written by end users as a concrete scenario, where such diversification is necessary and gives a benefit to the user.

On CNET, the end users are allowed to post their experiences of the use of the products and their opinions towards various features of the products in the form of reviews. The number of posted reviews may go up to hundreds in numbers for the popular products, where some reviews are more useful than others in addressing the pros and cons of the product under consideration. Thus, a set of reviews for a given product provides us with a comprehensive and detailed feedback about its useability experience. However, when the intention of a reader is to get a quick overview about all the pros and cons of various features of the product, reading all the reviews in a set can be a tedious and time consuming task. Hence, the challenge in this scenario is to come up with an optimal set of high quality reviews that cover as many relevant features as possible and provide diversified view points of opinions of different users about the product features.

### 1.2.1 Methodology

To come up with an optimal set of reviews as discussed in Section 1.2 poses certain challenges. The first challenge is to mine product features that are discussed in the review. The second challenge is to estimate the sentiments expressed by a user for various features in the review. And the third challenge is to come up with a strategy for selecting an optimal set of reviews that covers as many features as possible and associated diversified sentiments.

To confront the above mentioned challenges, we think the review selection problem as an information retrieval task with specific emphasis on result diversification and address it by combining latent semantic analysis with sentiment analysis. For modeling product features in the reviews we

---

<sup>6</sup><http://reviews.cnet.com/>

### 1.3. SCENARIO 3: MINING USER INTERESTS FROM SOCIAL CONTENTS<sup>5</sup>

use Latent Dirichlet Allocation and pre-process the review text using natural language processing techniques in a way that when used with LDA provides us with the topics that approximate product features discussed in the review. For estimating the overall review sentiment, we adopt a dictionary based approach and use the ANEW dictionary [8] which provides emotional ratings for a large number of English words. To select a subset of reviews that covers as many features as possible and diversified opinion range, we think of it as an optimization problem and formulate it as maximum coverage problem. It has already been shown that maximum coverage problem is NP-hard [120], therefore, we use a greedy approach for an approximate solution to the coverage problem. Empirical evaluation of our approach requires a test reference collection of reviews where each individual review is annotated for product features and associated sentiment in the review. To this end, we employed crowd sourcing approach [2] and developed a reference corpus of product reviews to be used in our evaluation. We use well established diversification evaluation measures to show that our FREuD approach performs better than baseline systems.

#### 1.2.2 Research Contribution

This part of our research provides the following contributions to the literature.

- We address the problem of social content diversification and develop the FREuD approach which combines machine learning algorithms with sentiment analysis techniques. Our FREuD approach provides a representative overview of sub-topics and aspects of discussions, characteristic user sentiments under different aspects, and reasons expressed by different opponents.
- To evaluate FREuD, we develop and contribute a novel test reference collection of product reviews which can be used for objective evaluation of various product review diversification algorithms. For this purpose, we use crowd sourcing to annotate reviews from various products for features and sentiment expressed for features in each of the reviews.

### 1.3 Scenario 3: Mining User Interests from Social Contents

Text contents are often categorized according to the subject matter or topic they address for better organization, grouping and understanding. A common observation is that a document may address one or more inter-related topics reflecting that the author is interested in more than one topic. Combining text analysis with author information does not only provide insight

into the topic structure of the documents but also helps to understand the topical interests of the authors. Adding time dimension to this analysis can further provide us an opportunity to inspect the evolution of topics and identify core authors at different stages of the topic life cycle. Hence, understanding topical trends and user roles in social media is an important challenge in the field of information retrieval.

For example, consider a scenario where a specific user tries to track a particular topic for its emergence, growth patterns, popularity and underlying key authors contributing to the topics over a period of time. In such a scenario manual analysis of this tremendous amount of text for finding topics, capturing topic evolution, identifying authors' interests and depicting changes in interests is expensive in terms of time and labor. Following this scenario the challenge is to provide a model which when used helps to capture topic evolution with authors' interests and roles in the context of evolving topics.

### 1.3.1 Methodology

To address the challenge mentioned in the previous section, we use machine learning techniques and apply Bayesian modeling of relations among authors, topics and temporal information. For this purpose, we extend state of the art Latent Dirichlet Allocation (LDA) [6] model and propose Author-Topic-Time (ATT) model. LDA is a probabilistic topic model which is extensively used for modeling topics, where a topic is seen as a group of observations that tend to co-occur more frequently than others. It assigns weights to the observations in the topic or group to depict the strength with which each observation belongs to that topic.

ATT is a three dimensional probabilistic model for jointly modeling the text, authors and time to capture topics, changes in users interests over time with respect to the evolving topics. In ATT, each topic is modeled as a distribution over words and each author is modeled as a distribution over topics. For modeling time, we use the Beta distribution. Exact inference in LDA type models is generally hard, therefore, we use the Markov chain Monte Carlo (MCMC) [77] approximation algorithm for learning the model parameters. For this purpose, we use the Gibbs sampler [36], which is a simple and special case of a Markov-chain Monte Carlo simulation and is particularly used for inference in high dimensional models. The most common way for measuring the performance of a topic model is to measure the predictive performance of the model using held out data or to use the model on some secondary task such as querying and classification. To test the performance of ATT, we run the model on a subset of abstracts of research papers collected from the CiteSeerX website and use perplexity to measure the predictive performance of ATT on held-out data and KL-divergence for measuring the quality of the topics determined by ATT.

### 1.3.2 Research Contribution

In this scenario our contribution is a novel probabilistic Author-Topic-Time (ATT) model for jointly modeling the text, author and time information in a probabilistic way. With our tests, we are able to show that ATT not only captures topic evolution but at the same also mines authors' interests which help categorizing authors with respect to their roles in the evolution of topics.

## 1.4 Thesis Structure

This thesis is divided into two parts. The first part consisting of Chapters 2 and 3, covers some basic and advanced concepts in Information Retrieval which are directly or indirectly related to our work. The second part which wraps Chapters 4 to 6, provides details of our research contributions using the scenarios given in Sections 1.1 to 1.3.

Chapter 2 is organized around the classical retrieval models and measures employed in the field of information retrieval. To this end, we provide details of the Boolean and Vector Space model with some thoughts on probabilistic retrieval models. Then we describe the performance measures employed for the evaluation of retrieval models. For this purpose, we cover details of two classical measures of retrieval performance i.e. Precision and Recall with other state of the art measures such as Mean Average Precision (MAP) and Normalized Cumulative Gain (nDCG) that are also used for measuring the performance of our approach given in Chapter 4. Later in the chapter we touch the topic of content diversification and briefly cover diversification performance measures such as  $\alpha$ -nDCG and Intent-Aware (IA) metrics, where  $\alpha$ -nDCG is used as a primary measure for our FREuD approach presented in Chapter 5.

In Chapter 3, we cover the fundamentals of probabilistic topic modeling. Chapter 3 takes the graphical approach to topic modeling and covers the representation, learning and inference techniques used for probabilistic models. We also cover the standard ways of measuring the performance of probabilistic models. To this end, we provide details of Perplexity and KullbackLeibler divergence measures which we also use in Chapter 6 for measuring the performance of our Author-Topic-Time (ATT) model.

Chapter 4 provides the details of our research contribution for Scenario 1.1. In the first half, we provide motivation to the problem of retrieval in Twitter and analyze the tweet features which contribute to the interestingness of a tweet. From this analysis, we come up with a method to use interestingness score of a tweet to predict the likelihood of a tweet to be retweeted. We also show that the interestingness score of a tweet can be used as a measure of static content quality for the tweets. In the second part of this chapter, we discuss in details the problem of sparsity and

document length normalization in Twitter and empirically show that our proposed measure of content quality can be effectively used to overcome the mentioned retrieval problems in Twitter. At the end of the chapter, we list the related work and provide a summary of our work.

In Chapter 5, we discuss the problem of social content diversification and propose the FREuD approach to overcome the content diversification problem stated in Scenario 1.2. We start the chapter with a motivational example and problem definition using CNET product reviews. Then, we show that diversification problem can be divided into three parts of aspect mining, sentiment estimation and review sub-set selection and provide a solution to each individual part to come up with the FREuD approach. In the next section, we elaborate on our strategy to objectively evaluate the performance of FREuD. To this end, we outline the approach used to develop the gold standard dataset and definition of the two competing baseline systems. In the end, we empirically show that our FREuD approach outperforms the two other baselines approaches in a diversification task. Lastly, we review the related work and end the chapter with a summary of our findings.

Chapter 6 details on our contribution mentioned in Scenario 1.3. In the chapter, we start with the introduction to the problem of mining topical trends and capturing author interests in the social media and provide a motivational example. Then, we move on to describe and define our proposed Author-Topic-Time (ATT) model solution for identifying latent topics, topic life cycle and author interests from the text contents. After defining the model, we elaborate on the parameter settings, application scenario and evaluation approach used for the ATT model. Later, we show the evaluation results of running the ATT model on CiteSeerX dataset. Towards the end of the chapter, we provide related work and conclude the chapter with the summary of our approach and findings.

We conclude our thesis in Chapter 7 and sum up our findings and contributions.

## 1.5 Dissemination

This section lists the contributions and publications that are based on the research work described in this thesis and are published in various conferences.

Our first contribution relates to the analysis and identification of tweet features that contribute to the interestingness of a tweet. Based on this analysis, we propose a static measure of content quality in Twitter. The findings of this work has been published in the paper “Bad News Travel Fast: A Content-based Analysis of Interestingness on Twitter” on 3<sup>rd</sup> *ACM Web Science Conference* in 2011 [82]. We extend this work and consider the problem of sparsity and document quality in Twitter. With the with

evaluation results, we show that our content quality measure proposed in paper [82] improves retrieval of interesting tweets. The results of this work has been published in the paper “Searching Microblogs: Coping with Sparsity and Document Quality” on 20<sup>th</sup> *ACM international conference on Information and knowledge management* in 2011 [84]. Further, we participated in TREC 2011 Microblog Track, where we use our approach from [82] to build a system and name it as LiveTweet. With LiveTweet results, we are able to show that our approach improves the result in filtering of interesting tweets. These results has been published in the paper “LiveTweet: Microblog Retrieval Based on Interestingness and an Adaptation of the Vector Space Model” on *Text Retrieval Conference (TREC)* in 2011 [16]. A demo of the LiveTweet<sup>7</sup> system is also presented on 34<sup>th</sup> *European conference on Advances in information retrieval* in 2012 [17].

Our second contribution relates to the problem of social content diversification. In this contribution, we propose the FREuD approach for sentiment based diversification of product reviews. An early version of this approach has been published in a short paper ‘FREuD: Feature-Centric Sentiment Diversification of Online Discussions’ on 4<sup>th</sup> *ACM Web Science Conference* in 2012 [85]. We extend the FREuD approach from the paper [85] and develop a gold standard dataset for thorough and sound evaluation. With the evaluation results from a real world dataset, we are able to show that FREuD outperformed the baseline systems. The results of this work are published in the paper ‘Feature Sentiment Diversification of User Generated Reviews: The FREuD Approach’ on 7<sup>th</sup> *International AAAI Conference on Weblogs and Social Media* in 2013 [86].

Our final contribution in this thesis is the Author-Topic-Time model which captures topical trends and authors’ interest in the social media. We publish an early idea of this approach in a short paper “ATTention: Understanding Authors and Topics in Context of Temporal Evolution” on 33<sup>rd</sup> *European conference on Advances in information retrieval* in 2011 [88]. Later, we extend this work and evaluate our model on real world dataset collected from CiteSeerX and these results has been published in the paper “ATT: Analyzing Temporal Dynamics of Topics and Authors in Social Media” on 3<sup>rd</sup> *ACM Web Science Conference* in 2011 [87].

---

<sup>7</sup><http://livetweet.west.uni-koblenz.de/>





## Chapter 2

# Foundations of Information Retrieval

Today's era of internet has made it easy for the people to contribute and share information with other people. The vast amount of this information is available in different formats such as text, images, videos etc. and in most cases is freely made accessible to others online in a variety of ways. Information overload is one of the main problems which is prevalent now a days and is faced by many information seekers. The Information Retrieval (IR) discipline [52, 110] born about half a century ago, is one area of research which aims at solving this type of problem. This discipline deals with the representation, storage, organization of, and access to unstructured information items. Classically information retrieval focuses on retrieving information items which are usually available in free-form natural language text. The aim of information retrieval is to build computer systems which allow users to express their "*information needs*" in order to retrieve "*information*" which is relevant to their needs.

The focus of this chapter is to provide a brief overview about evolution of information retrieval models, common measures available for measuring their performance and some applied information retrieval topics.

The rest of this chapter is organized as follows. Section 2.1 provides a glimpse of IR history and covers three fundamental models used in information retrieval. Evaluation measures for retrieval models are discussed in Section 2.2. We will also touch the topic of search results diversification along with diversification metrics in Section 2.3. In the end, we summarize our discussion about models and measures presented in this chapter.

### 2.1 Information Retrieval Models: an Overview

In 1945, Vannevar Bush published an article [10] "As We May Think", in which he described the idea of using computers for automatic access

to large amount of stored knowledge. During 1950's this idea materialized into more formal methods of automatically searching text collections using library classification schemes. H.P. Luhn in 1957 [75] put forth an information retrieval criteria based on word overlap using words as indexing units for documents. The most noticeable development in 1960's were the start of Cranfield Projects [24, 25] by Cyril Cleverdon at Cranfield and the SMART system [109] from Gerard Salton at Havard which allowed researchers to conduct experiments in order to improve search quality. The Cranfield project proposed experiments to test the effectiveness of four indexing schemes prevalent at that time for organizing information. The experts from each scheme were invited to participate in the experiments and the task was to index document, design search strategies and perform search operations. The results of these experiments provoked further debate which led to the start of second Cranfield project. The most important outcome of Cranfield 2 was the definition of different notions of methodology for information retrieval experiments, the details of which can be found in [24, 25]. The SMART system [109] project resulted in many of the ideas that are now a days implemented in search engines, for example, a notion of scoring function for measuring relevance and consequent ranking of documents for display to user.

The era of 1970s'and 1980s' saw the development of various information retrieval models which were effective on small text collections. The lack of large test collections at that time hindered researches to test those models in large scale experiments. However in 1992 with the onset of Text Retrieval Conference (TREC) [45] large text collections were made available and researchers were able to test the old information retrieval models in large scale experiments.

### 2.1.1 Formal definition of an IR Model

As defined by Ricardo Baeza-Yates in [4] "An *information retrieval* model is a quadruple  $[D, Q, F, R(q_i, d_j)]$  where

- $D$  is a set composed of logical views (or representations) for the documents in the collection.
- $Q$  is a set composed of logical views (or representations) for the user information needs. Such representations are called queries.
- $F$  is a framework for modeling document representations, queries and their relationship.
- $R(q_i, d_j)$  is a ranking function which associate a real number with a query  $q_i \in Q$  and a document representation  $d_j \in D$ . Such ranking defines an ordering among the documents with regard to the query  $q_i$ "

Retrieval models can be categorized on the basis of either their mathematical basis or on the basis of their properties. Figure 2.1 shows the relationship of common retrieval models based on their category and properties.

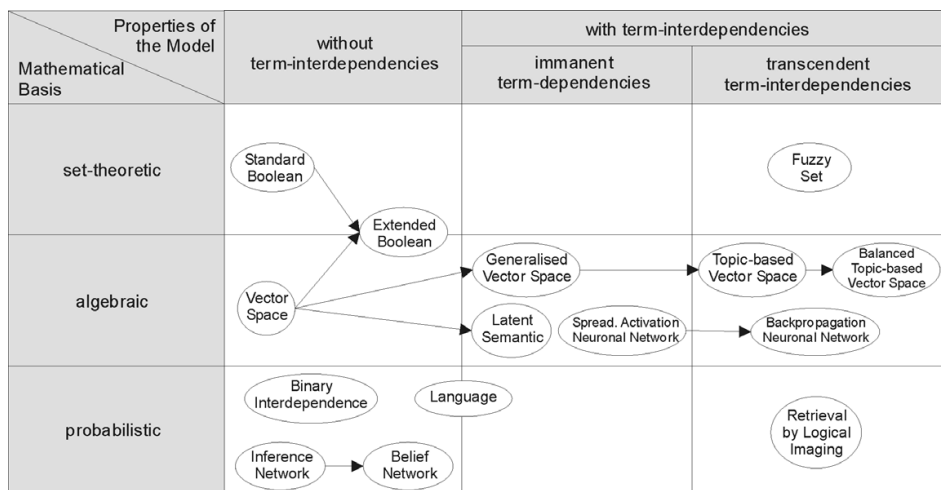


Figure 2.1: Information retrieval models categorized on their mathematical basis and properties. (Source: [68])

Before discussing models, an important concept in IR is *indexing*. Indexing corresponds to describing documents by set of keywords called index terms. Index terms are the words in a document which are distinct and convey the main theme of a document. There exist criteria that help in finding which terms are more useful than others in a document to be used as index terms. For example, one criterion is to use those document words as index terms that are distinct for the document given all other words in a document collection.

Due to limited scope of this chapter, we limit our discussion to two fundamental retrieval models which form the basis for other more advanced models. In the next section, we cover these two models in detail.

### 2.1.2 Boolean Model

The Boolean model is based on set theory and Boolean algebra. It views documents as a set of index terms and user information need as Boolean expression on index terms. It considers index terms either present or absent in the document and weighs them accordingly as one (if term is present) or zero (if term is absent). In a Boolean query, the terms are combined using classical Boolean operators **AND** ( $\wedge$ ), **OR** ( $\vee$ ) and **Not** ( $\neg$ ). A document is considered relevant to a query iff it satisfies the query expression. For example, if a Boolean query  $q$  is formulated as  $(t_a \wedge t_b)$  then only those

documents are considered relevant which contain both of the query terms in them. A Boolean model can not find a partial match and can only predict a document as either relevant or non-relevant. One can build complex queries by using any number of Boolean operators in any combination and are evaluated according to Boolean algebra rules. An important refinement of Boolean queries is the use of “proximity“ operator with the standard Boolean operators. This operator specifies the spatial distance between two query terms with in the document. The units of the distance can be word, sentence, paragraph etc. The proximity operator may also be used to specify the order of the query terms.

The main advantage of the Boolean model lie in its‘ clean formalization which is easy to understand and implement. The disadvantages of the Boolean model includes retrieval of exact matches that results in retrieval of either too few or too many documents, no notion of partial matches, no ranking of retrieved documents, it can not use term weights and sees all terms as of equal importance.

### **2.1.3 Vector Space Model**

In the vector space model, a document is represented as a weighted vector referred to as term vector. The terms in the term vector may represent any entity of interest. For text documents the terms are the words extracted from the document text. To prepare a term vector, we extract words from documents, remove stop words and in many cases the words are stemmed as well. The stop words are the words which frequently occur in documents such as “a”, “the”, “to” etc. and have little discriminatory power. Stemming refers to reducing morphological variants of words to their stem or root. It does not only help in obtaining a single representation of different variants of a word but also reduces the number of distinct terms required to represent a set of documents, which in turn reduces the processing time and amount of space required for storing term vectors. Each term in the document collection represents one dimension and the set of all terms define a space. As each document is represented by a set of terms, we can view this space as document space.

As opposed to the Boolean model, the vector space model assigns non-binary weight to each of the term in a document referred to as term weight. Each term may be assigned a different weight in different documents estimating the usefulness of the term in distinguishing the given document from others in the same document collection. We can represent a document as a point with term weights assigned to the terms as coordinates of a document in the document space. More formally, each document is interpreted as a vector from the origin in document space to the point defined by the documents‘ coordinates.

The document space can also be viewed as a document-term matrix,

where each row of the matrix constitutes a document. The entry at  $i$ th column and  $j$ th row is the weight of the term  $i$  in document  $j$ . Similarly, we can also represent the users' information need or query as a weighted vector of term in the document space.

Once we are done with how to represent both the documents and query in the document space, the next question is the computation of term weights. Different term weighing schemes are used for this purpose which are discussed in detail in [112]. The most common and widely used is *tf-idf* scheme. A "term frequency" (*tf*) is the count of a specific term  $w$  in a given document. It provides an estimate of the importance of a term in a document. It is a document specific statistics and varies from document to document for a given term. We normalize the *tf* to counter bias towards the longer documents using the following

$$tf_{i,j} = \frac{\text{freq}(i,j)}{\max\{\text{freq}(w,j) : w \in j\}} \quad (2.1)$$

The maximum is calculated over all the terms and is the frequency of the term that occurs most frequently in the document  $j$ .

The "inverse document frequency" (*idf*) characterizes a given term with in the entire document collection. It is a global statistic and is a measure of how widely a term is distributed in a document collection. The *idf* is defined as

$$idf_i = \log \frac{N}{n_i} \quad (2.2)$$

Where  $N$  is the total number of documents in a collection and  $n_i$  is the number of documents containing term  $i$ . If a term appears in every document then its *idf* is zero signaling that this term has no importance in distinguishing a relevant document from a non-relevant document. After computing *tf* and *idf* for a given term, the weight of a term in a given document is computed by

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{n_i} \quad (2.3)$$

or by some other variations of Equation 2.3. The term weighing schemes which use *tf-idf* as basis for computing term weights are called *tf-idf* schemes.

After computing the vectors for the query and documents in the collection using some *tf-idf* weighting scheme, in the next step we compute the degree of similarity between query and each document. The documents are then ranked in the decreasing order of similarity to the query. As vector space model provides numeric scores of similarity therefore it takes into account the partial matches. As the documents and query can be represented as vectors in the document space, the usual similarity measure then is to use the cosine of the angle between the query vector and the document vector

as shown in Figure 2.2. The cosine similarity between document vector  $j$  and query vector  $q$  is calculated by computing the inner product of these two vectors and is given as

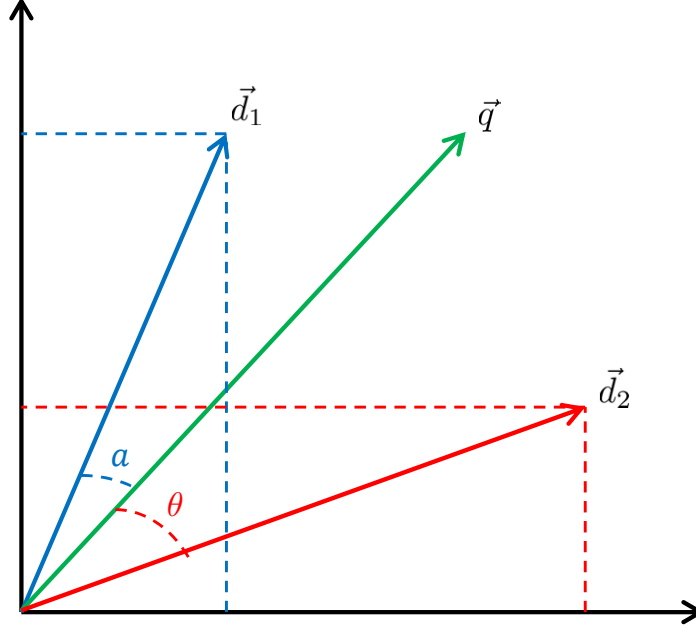


Figure 2.2: The cosine of  $\theta$  is adopted as  $sim(d, q)$ . (Source: [127])

$$sim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{\|\vec{d}_j\| \|\vec{q}\|} = \frac{\sum_{i=1}^t w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^t w_{i,j}^2} \times \sqrt{\sum_{j=1}^t w_{i,q}^2}} \quad (2.4)$$

Where  $\|\vec{d}_j\|$  and  $\|\vec{q}\|$  are the norms of the document and query vector, which for a given vector is computed as follows

$$\|\vec{q}\| = \sqrt{\sum_{i=1}^t q_i^2} \quad (2.5)$$

We obtain the maximum similarity of 1 when both document and query vectors are identical and cosine similarity of zero when both vectors are orthogonal to each other. The main advantages of the vector space model are its ability to retrieve partial matches by using the term-weighting schemes and sorting the documents using cosine similarity scores with respect to the given query. The disadvantages of the vector model are: its inability to map the term dependencies, long documents result in poor similarity values and that it only allows exact match of search and document terms.

### 2.1.4 Extended Boolean Model

To overcome the problem of exact match in Boolean model, Gerard Salton in [111] proposed Extended Boolean model. It combines the Boolean model with ranking capability of vector space model. The document in Extended Boolean model is represented as a weighted vector of terms. Users are allowed to use **AND** and **OR** operators as in the Boolean model but the keywords are weighted and documents are ranked by a similarity function. The weight of a term  $x$  in document  $j$  is computed using its normalized term frequency as given below.

$$w_{x,j} = tf_{x,j} \cdot \frac{idf_x}{\max_i idf_i} \quad (2.6)$$

where  $tf_{x,j}$  is the normalized frequency of term  $x$  in document  $j$ ,  $idf_x$  is the inverse document frequency of the term  $x$  and  $idf_i$  is the inverse document frequency of the term  $i$  that has maximum frequency in document  $j$ .

### 2.1.5 Probabilistic Models

The probabilistic method to retrieval is based on the general principle that documents in a collection should be assigned a probability with which they are relevant to a query and ranking should be based on the decreasing order of probability relevance. Probabilistic models should be distinguished from other models such as vector space model where ranking is based on the similarity measure whose values are not directly interpretable as probabilities. Cooper et al. [26] sums up the potential advantages of probabilistic approach as :

- “One has grounds for expecting retrieval effectiveness that is near-optimal relative to the evidence used”.
- There should be “less exclusive reliance on traditional trial-and-error retrieval experiments ... to discover the parameter values that result in best performance”. Different term weighing schemes used in vector space model are example of such trial and error experiments.
- “An array of more powerful statistical indicators of predictivity and goodness of fit [than precision, recall, etc.] become available”.
- “Each documents’ probability-of-relevance estimate can be reported to the user in ranked output ... It would presumably be easier for most users to understand and base their stopping behavior [i.e., when they stop looking at lower ranking documents] upon ... a ‘probability of relevance’ than [a cosine similarity value]”.

The true or actual probabilities in IR model are hard to compute analytically (c.f. Chapter 3), therefore, probabilistic models try to *estimate* the

probability of relevance of a document to the query. This estimation is the key part of the model, and serves as basis for differentiating one probabilistic model from other. The fundamental idea in probabilistic retrieval models is that for a given query  $q$  there is a set  $\mathcal{R}$  which contains the relevant documents and the complement of this set is  $\bar{\mathcal{R}}$  which contains non-relevant documents.  $p(\mathcal{R}|d)$  denotes the probability that a document is relevant to the query and  $p(\bar{\mathcal{R}}|d)$  is the probability that document is not relevant. The similarity of the document to the query is defined as follows

$$sim(d, q) = \log \frac{p(\mathcal{R}|d)}{p(\bar{\mathcal{R}}|d)} \quad (2.7)$$

Using Bayes' rule Equation 2.7 becomes

$$sim(d, q) = \log \frac{p(d|\mathcal{R}) \cdot p(\mathcal{R})}{p(d|\bar{\mathcal{R}}) \cdot p(\bar{\mathcal{R}})} \quad (2.8)$$

$p(d|\mathcal{R})$  is the probability of randomly selecting the document  $d$  from the set  $\mathcal{R}$  of relevant documents and  $p(d|\bar{\mathcal{R}})$  is the probability of randomly selecting the document  $d$  from the set  $\bar{\mathcal{R}}$  of non-relevant documents.  $p(\mathcal{R})$  is the prior probability that a randomly selected document is relevant to the query and is constant across the collection and same is true for  $p(\bar{\mathcal{R}})$ . Since  $p(\mathcal{R})$  and  $p(\bar{\mathcal{R}})$  are just scaling factors and thus can be removed from above formulation leaving us with simplified equation

$$sim(d, q) = \log \frac{p(d|\mathcal{R})}{p(d|\bar{\mathcal{R}})} \quad (2.9)$$

In the simplest form of this model we assume that terms are mutually independent of each other (*independence assumption*) and  $p(d|\mathcal{R})$  can be written as joint probability of individual term probabilities, i.e. the probability of presence/absence of a term in relevant/non-relevant documents and is given by

$$p(d|\mathcal{R}) = \prod_{t_i \in q, d} p(t_i|\mathcal{R}) \times \prod_{\bar{t}_i \in q, d} (1 - p(\bar{t}_i|\mathcal{R})) \quad (2.10)$$

Where the first factor of the product in above equation uses probability of presence of a term  $t_i$  in a document randomly selected from the set  $\mathcal{R}$  of relevant documents, and second factor uses the probability of absence of a term  $\bar{t}_i$  from a document randomly selected from set  $\mathcal{R}$ . The denominator in Equation 2.9 can also be defined in the similar way. Using Equation 2.10 we can rewrite  $sim(d, q)$  as follows

$$sim(d, q) = \log \frac{\prod_{t_i \in q, d} p(t_i|\mathcal{R}) \times \prod_{\bar{t}_j \in q, d} (1 - p(\bar{t}_j|\mathcal{R}))}{\prod_{t_i \in q, d} p(t_i|\bar{\mathcal{R}}) \times \prod_{\bar{t}_i \in q, d} (1 - p(\bar{t}_i|\bar{\mathcal{R}}))} \quad (2.11)$$



Recalling that  $p(t_i|\mathcal{R}) + p(\bar{t}_i|\mathcal{R}) = 1$  and using only the terms present in the document, we can rewrite the Equation 2.11 as

$$sim(d, q) = \sum_{t_i \in q, d} \log \frac{p(t_i|\mathcal{R})}{1 - p(t_i|\mathcal{R})} + \log \frac{1 - p(t_i|\bar{\mathcal{R}})}{p(t_i|\bar{\mathcal{R}})} \quad (2.12)$$

The probabilities  $p(t_i|\mathcal{R})$  and  $p(t_i|\bar{\mathcal{R}})$  can be computed in various ways and one such method as given in [27] is to assume that  $p(t_i|\mathcal{R})$  is constant for all the terms and typically is set to 0.5. It is also assumed that for a given query almost all documents are non-relevant and  $p(t_i|\bar{\mathcal{R}})$  can be estimated by  $\frac{n_i}{N}$ , where  $N$  is number of documents in the collection and  $n_i$  are the number of documents containing term  $i$ . Using these assumptions the scoring function can be written as

$$\sum_{t_i \in q, d} \log \frac{N - n_i}{n_i} \quad (2.13)$$

which is similar to the inverse document frequency function discussed in Section 2.1.3.

The main disadvantages of probabilistic methods are that the term weights are assumed as binary and do not consider the frequency of a term in the document and that they also assume that terms are independent of each other.

## 2.2 Evaluation Measures

Evaluation of a software system can be carried out in a number of ways. Selecting the evaluation type for measuring the performance of a system depends on the objective of the system. The most common types of evaluations are functional and performance evaluations. In the context of information retrieval systems the retrieval performance of the system is also measured. Functional performance includes checking if the system provides functions for which it is designed, while performance evaluation includes measuring the time and space required to complete the intended function of the system. Since IR systems are designed to satisfy human information needs, other than measuring time and space we are also required to measure how good is an IR system in providing the required information. In a traditional scenario for a retrieval system, a human subject issues a query and the system returns a set of documents that best match the query topic. An IR system should not only provide relevant documents but also rank them in a decreasing order of relevance to the query. Hence, measuring the performance of an IR system requires measuring the relevance of the retrieved documents to a given query. Such an evaluation is usually based on a ground truth or test reference dataset and on evaluation measures. The ground truth data

set consists of a set of documents that are assessed as relevant by human subjects for a given information need in a document collection. For a given information need or query, evaluation metrics are used to measure the similarity between the set of documents returned by the retrieval system and documents in the ground truth data set.

Objective evaluation of retrieval performance has been the corner stone of IR. Since the inception of IR field, it was evident in the IR community that objective evaluation of retrieval performance will play a key role in the field. The Cranfield experiments conducted in 1960's, established the desired set of characteristics for a retrieval system. The two classical properties of an IR system that are widely accepted and used by the IR community for measuring retrieval performance is *Precision* and *Recall*.

### 2.2.1 Precision and Recall

Consider an example where we have a test reference collection  $\mathcal{R}$  of documents that are related to a set of queries  $\mathcal{Q}$ . For a given query  $q$ , the retrieval system returns a set of documents  $\mathcal{A}_q$ , and  $\mathcal{R}_q$  is the set of documents that are relevant to the query in the collection  $\mathcal{R}$ . Then the precision and recall is defined as:

- **Precision** denoted by  $P$  is defined as the fraction of the retrieved documents that are relevant to the query  $q$

$$P = \frac{|\mathcal{R}_q \cap \mathcal{A}_q|}{|\mathcal{A}_q|} \quad (2.14)$$

- **Recall** denoted by  $R$  is defined as the fraction of the relevant documents that are successfully retrieved.

$$R = \frac{|\mathcal{R}_q \cap \mathcal{A}_q|}{|\mathcal{R}_q|} \quad (2.15)$$

The goal of an IR system should be to maximize the precision and recall. However, these two goals are opposite to each other. Retrieval algorithms that improve recall tend to hurt precision and vice-versa. Precision takes in to account all the documents that are returned by the IR system, however it can also be measured at a given cut-off rank by considering the top most relevant documents. This number is reported as  $P@k$ , where  $k$  is the cut-off rank or recall level in the result set and usually set at 0%, 10%, 20%,  $\dots$ , 100%.

### 2.2.2 Average Precision

Both recall and precision are set oriented measures and have no notion of ranked retrieval. Precision and recall are computed on the basis of the whole

list of documents returned by the system and do not consider the ranking order of the documents. For a retrieval systems which returns a ranked list of documents, average precision is a more appropriate measure of the systems' performance. The idea is to first calculate the precision and recall at each position in the ranked result set and then compute the average precision with respect to the number of relevant documents in the result set as given in Equation 2.16.

$$AveP_q = \frac{\sum_{k=1}^n (P(k) \times rel(k))}{|\mathcal{A}_q|} \quad (2.16)$$

Where  $k$  is the rank or position of document in the resultset,  $P(k)$  is the precision at cut-off rank  $k$  in the resultset and  $rel(k)$  is the binary relevance of document at rank  $k$ . This measure favors systems which produce more relevant documents at the top of the result set.

### 2.2.3 Mean Average Precision (*MAP*)

The Equations (2.14) to (2.16) provide precision and recall estimates for a single query. In most of the evaluations, the retrieval algorithms are run for multiple distinct queries and performance is measured over the complete set of queries  $\mathcal{Q}$ . For this purpose, we calculate average precision for each individual query as given in Equation 2.16 and then compute *MAP* as given in Equation 2.17, which is a mean of the average precision values for all the queries in set  $\mathcal{Q}$ .

$$MAP = \frac{\sum_{q=1}^{|\mathcal{Q}|} AveP(q)}{|\mathcal{Q}|} \quad (2.17)$$

### 2.2.4 Normalized Discounted Cumulative Gain (*nDCG*)

The performance measures we have discussed so far assume that a document is either relevant or not relevant for a given query. However, in many scenarios the relevance of a document for a given query is judged on a graded scale of relevance. The intuition is that not all documents equally satisfy human information need. Some documents are more useful than others that still provide some information regarding the query. The idea is to mark a document on a relevance scale of 0 to 3 (the range of the scale may vary depending upon the concrete experimental setup), where 0 means irrelevant, 1 means marginally relevant, 2 means relevant and 3 means that a document is highly relevant to the query. This relevance judgment is subjective in nature and may vary from one test subject to another. Once we have graded relevance judgment for documents, in the next step we use **Cumulative Gain** (*CG*) to compute the gain vector. *CG* measures the overall gain provided by a result set without considering the position of the document in

the result set. It is just the sum of graded relevance value for each document in the result set and is given by

$$CG_k = \sum_{i=1}^k rel(i) \quad (2.18)$$

Where  $rel(i)$  is the graded relevance value for the document at position  $i$  in the result set. As any change in the order of the documents in the result set will not affect the  $CG$ , that is why  $CG$  can not determine which system is better if both return the same document set but in different order of graded relevance values. It is assumed that a retrieval system ranks the documents in a decreasing order of relevance starting at highly relevant, relevant, marginally relevant and then non-relevant at the end of result set.  $DCG$  takes care of this order and discounts the gain of a document if it appears lower in the order but is more relevant than others which appear higher in the order. The  $DCG$  for a result set at cut-off rank  $k$  is calculated using the following equation

$$DCG_k = rel_1 + \sum_{i=2}^k \frac{rel(i)}{\log_2(i)} \quad (2.19)$$

Since, a retrieval system can produce result sets of different lengths for different queries or two retrieval systems can produce result sets of different length for the same query, therefore,  $DCG$  values are not helpful for comparing a system across different queries or comparing two different retrieval systems. For this comparison, we normalize the  $DCG$  score for each query using the **Ideal Discounted Cumulative Gain** ( $IDCG$ ) at cut-off rank  $k$ .  $IDCG$  is produced by first sorting the result set in decreasing order of graded relevance and then computing  $DCG$  for this ranking till the  $k$ th position in the result set. The normalized  $DCG$  is given by

$$nDCG_k = \frac{DCG_k}{IDCG_k} \quad (2.20)$$

Further details of  $nDCG$  can be found in [55]. To measure the average performance of a retrieval system, one can take the mean of  $nDCG$  values for all the queries.

## 2.3 Content Diversification

So far our discussion in previous sections pertain to measuring the performance of a retrieval systems where the goal is to retrieve as many as possible documents which are relevant to the user information need. In many real world cases the queries issued by the users are either ambiguous or underspecified. An ambiguous query has multiple distinct interpretations.

Consider an example where the user issues an ambiguous query “Apple“, this may mean user is either interested in the apple fruit or the Apple company. An underspecified query is one which has multiple aspects or intents. For example, the query “iPhone“ is an underspecified query where a user is interested in knowing about various features or aspects of the phone. Of course, at the same time a query may be both ambiguous and underspecified. In such scenarios, the goal of the retrieval system should be to return a ranked and diversified result set that covers both the breadth of available information and all possible intents of the query.

Search result diversification is an optimization problem aiming to find a subset of  $k$  documents containing both most relevant and most diverse information. In [11, 19], it has already been shown that diversification requires a trade-off between having more relevant results pertaining to the correct intent and having more diverse results at the top of the result set for a given query. Therefore, any retrieval system which optimizes relevance and diversity must find a way to establish a balance between both the competing objectives. Finding the best solution for this type of optimization problem is NP-hard [120]. The solution is to use either approximation algorithms or some greedy approach.

Further discussion about content diversification is provided in Chapter 5.

### 2.3.1 Diversity Evaluation Measures

The evaluation measures discussed in Section 2.2 assume that relevance judgment of each document in the result set can be done in isolation, independently of other documents. This type of assumption leaves space for duplicate or near duplicate documents to be the part of result set. A retrieval system which attempts directly or indirectly to use these measures for optimizing its objective function may achieve high score on standard evaluation measures but can produce unsatisfactory results when the user requires diverse and novel information for a given query. In such cases an appropriate measure would be the one which combines both relevance and diversity together in evaluation.

In the past few years several diversity evaluation measures have been proposed which attempt to combine relevance and diversity in ranked document retrieval. Examples of such measures are  $\alpha$ - $nDCG$  [22], *Intent-Aware* metrics [1] and the  $D\#$ -measure (Dee Sharp) [106]. Below we briefly discuss two of these measure.

### 2.3.2 $\alpha$ - $nDCG$

The  $\alpha$ - $nDCG$  measure was proposed by Clarke et al. [22] incorporates both novelty and diversity for measuring the effectiveness of a retrieval system

in diversity ranking. It is based on the notion of *nugget* which is seen as a binary property of the document representing an aspect, intent or subtopic of a given query. They modeled the users' information needs as sets of nuggets. The  $\alpha$ -*nDCG* defines graded relevance as the number of different nuggets covered in each document. For  $\alpha$ -*nDCG* a highly relevant document is one which covers many nuggets or intents. It discounts a document if it covers a nugget that has already been covered and then the document is further discounted on the basis of its rank as given in *nDCG*. If a retrieval system returns two documents that cover the same nugget the second document is deemed as redundant and will be discounted.

The assumption underlying  $\alpha$ -*nDCG* is that each query has multiple known intents or facets and these intents are of equal importance. The  $\alpha$ -*nDCG* metric regards the documents in a result set to cover these query intents to different degrees. A highly relevant document is one which covers many intents. Additionally,  $\alpha$ -*nDCG* promotes an increase in diversity by reducing redundancy.

The main difference between standard *nDCG* and  $\alpha$ -*nDCG* lies in the definition of the gain values. For  $\alpha$ -*nDCG*, the gain  $G[k]$  of the document at rank  $k$  is a vector over all query intents. Furthermore, this gain is discounted for intents which have already been covered by higher ranked documents. Thus, for  $\alpha$ -*nDCG*, the gain  $G[k]$  at rank  $k$  is defined as given in Equation 2.21

$$G[k] = \sum_{i=1}^m J(d_k, i)(1 - \alpha)^{r_{i,k-1}} \quad (2.21)$$

where  $J(d_k, i)$  is a binary value describing if the document at rank  $k$  is relevant to the query intent  $i$  according to the gold standard and  $r_{i,k-1}$  denotes how many higher ranked documents have already addressed the intent  $i$ . The parameter  $\alpha$  is used to balance redundancy and novelty. Its value ranges between 0 and 1, where lower values of  $\alpha$  increase redundancy and decrease novelty and higher values of  $\alpha$  favour novelty at the cost of reduced redundancy.

Based on these gain values the discounted cumulative gain *DCG* [ $k$ ] at rank  $k$  is given below in Equation 2.22:

$$DCG[k] = \sum_{j=1}^k G[j] / (\log_2(1 + j)) \quad (2.22)$$

Defining  $DCG'$  as the ideal gain obtained by sorting the documents in the ideal order according to the gold standard allows for a normalization equivalently to the one of standard *nDCG*. This leads to the definition of  $\alpha$ -*nDCG* [ $k$ ] as below in Equation 2.23

$$\alpha - nDCG [k] = \frac{DCG [k]}{DCG' [k]} \quad (2.23)$$

### 2.3.3 Intent Aware (IA) Metrics

Consider the example of our query *Apple*, which may mean either the user is interested in intent *fruit* or intent *company*. If we know the distribution of query intents, we can use this information for ranking the documents so that result set contain more documents that are relevant to query intent that has higher likelihood. If the query intents are known in advance, one can use this information to obtain per-intent graded relevance assessments. The measures which incorporate intent likelihood in evaluation are called *Intent-Aware* metrics [1]. The examples of such measures include *NDCG-IA*, *MAP-IA* and *MRR-IA* etc.

With the classical *nDCG*, we compute the ratio of *DCG* to *IDCG* at a given cut-off rank. However, when a query has multiple intents of varying importance then *DCG* and *IDCG* both depend on the intents the results are evaluated against. Given the distribution on the  $n$  intents of a query and per-intent graded relevance judgment, we can compute the *DCG-IA* for a given ordering of results at cut-off rank  $k$  as below

$$DCG-IA [k] = \sum_{i=1}^n p(i) \sum_{j=1}^k \frac{rel(i, j)}{\log_2(j+1)} \quad (2.24)$$

Where  $p(i)$  is the probability of the intent  $i$  for a given query and  $rel(i, j)$  is the relevance of  $j^{th}$  document with respect to the  $i^{th}$  intent.

## 2.4 Summary

This chapter presented a brief overview of the history and developments made in the IR field over the past years. We described the principles underlying the Boolean, Vector Space, and Probabilistic models of information retrieval and provided an overview of the classical and few other state of the art retrieval measures. In the end we touched the topic of search result diversification with some metrics that are employed in diversification ranking.





## Chapter 3

# Foundations of Probabilistic Modeling

We mentioned in Chapter 1 the amount and speed at which the contents are generated in the web. Most of the tools and techniques of today are not yet capable to exploit this big amount of human generated data. We need new techniques and tools which can enable us in organizing, searching, and understanding this huge amount of data collections. One area of research which deals with this kind of information management is topic modeling. Techniques and methods provided by topic modeling have the potential for automatically organizing, understanding, searching and summarizing large collections of data. We can use topic modeling to uncover the latent topical patterns that dominate a collection. We can annotate these collections with the discovered latent topics and later can use these annotations to organize, summarize and search the texts. In topic modeling a 'topic' is seen as a group of observations that tend to co-occur frequently and a document or a collection of documents can be described or explained using these topic(s).

Topic modeling research find its roots in directed graphical models, hierarchical Bayesian methods, conjugate and non conjugate priors, modeling with graphs, approximate posterior inference, model selection, exploratory data analysis, and nonparametric Bayesian methods. In this chapter, we provide an overview of the representation, learning and inference techniques that are prerequisite for understanding the way topic modeling works.

The rest of this chapter is organized as follows. In Section 3.1, we present an overview of the graphical models with specific emphasis on Bayesian network. Section 3.3 describes the techniques that are commonly used for estimating parameters in graphical models. In Section 3.4, we provide an overview of approximate inference methods used in topic models and Section 3.5 describes the ways for evaluating topic models.

### 3.1 Graphical Models

Many of the tasks in real world require reasoning by humans or by machines (automated systems) under uncertainty to arrive at some decision or conclusion given some observations. Graphical models [57, 58, 71, 94] provide a general framework and approach for modeling uncertainty through the use of probability theory for such kind of tasks. We can characterize complex systems using multiple interrelated aspects and can use these aspects for reasoning. For example, in a document generation system a document is generated using a language model over some *vocabulary*. The generated document contains information of some *kind* in it. We can model these aspects using random variables and the value of each variable can be used to define a property of the document generation system. In such a system, our task is confined to probabilistic reasoning about the latent values of variables given some observed variables. For reasoning, we may construct joint distribution over the random variables  $\mathcal{X} = \{X_1, \dots, X_n\}$  in the system. For a simple case, if these are the binary variables then joint distribution requires the specification of  $2^n$  numbers. As the number of variables and set of possible values increase in the system, the specification of joint distribution becomes extremely complex.

Probabilistic graphical models provide a framework to compactly represent the complex joint distributions. This framework uses graph-based methods for representing and encoding complex distributions over high dimensional space. The advantages provided by graphical framework are,

- **Representation:** In case of large joint distribution it allows to represent the distributions tractably such that humans can also understand and evaluate its semantics and properties.
- **Learning:** The graphical framework provides an effective way for model construction that provides a good approximation of the past experience by learning from data.
- **Inference:** Posterior distributions can be computed from the prior distributions using the same structure. Inference algorithms runs on the graph structure and are faster than computing the joint distributions explicitly.

The two common classes of graphical models that are used for describing discriminative and (or) generative probabilistic models are Bayesian networks (directed graphical models) and Markov networks (undirected graphical models). Bayesian networks are used to show the causal relationship between random variables, while Markov network are better at expressing soft constraints between random variables. For inference problems, both directed and undirected graphs are changed to a different representation know as factor graphs.

### 3.2 Bayesian Networks (Representation)

The Bayesian network is a directed acyclic graph where nodes correspond to random variables and edges denote dependencies between the random variables. Lets take an example of three random variables  $x, y$  and  $z$ . To show the use of Bayesian network to describe probability distributions, we can consider a joint distribution  $p(x, y, z)$  over these variables. Using product rule of probability, we can write the joint distribution of  $x, y$  and  $z$  as

$$p(x, y, z) = p(x) p(y|x) p(z|x, y) \quad (3.1)$$

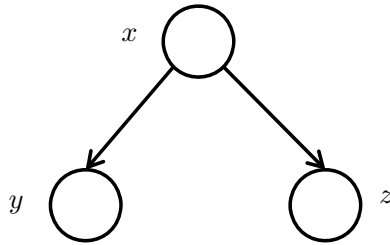


Figure 3.1: A directed graphical model showing the joint probability distribution over three variables  $x, y$ , and  $z$  reflecting to decomposition in the right hand side of Equation 3.1. (Source: [5], p. 361)

The fully connected graphical model which corresponds to this joint distribution is shown in Figure 3.1 and defines the pattern of conditional dependence of the random variables with arrows showing the directionality of the dependence. Each variable is represented by a node in the graph with corresponding conditional distribution on the right-hand side of the Equation 3.1. Directed links are added to the graph from the nodes corresponding to the variables on which the distribution is conditioned. The direction of the arrow shows the parent-child relationship between nodes. We can extend the graph by adding any number of nodes with each node showing one random variable. For example a joint distribution over  $N$  random variables using product rule of probability can be written as a product of conditional distributions as given in Equation 3.2.

$$p(t_1, \dots, t_N) = p(t_1) p(t_2|t_1) \dots p(t_N|t_1, \dots, t_{N-1}) \quad (3.2)$$

The above factorization is of a fully connected graph, however there exists situations when the graph is not fully connected as shown in Figure 3.2.

The joint distribution corresponding to the nodes in this graph as product of a set of conditional distributions one for each node conditioned only on its parents is given in Equation 3.3.

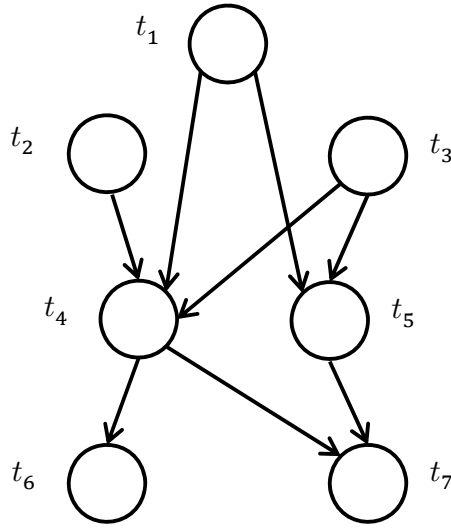


Figure 3.2: Example of a directed acyclic graph describing the joint distribution over variables  $t_1, \dots, t_7$ . The corresponding decomposition of the joint distribution is given by Equation 3.3. (Source: [5], p. 362)

$$p(t_1) p(t_2) p(t_3) p(t_4|t_1, t_2, t_3) p(t_5|t_1, t_3) p(t_6|t_4) p(t_7|t_4, t_5) \quad (3.3)$$

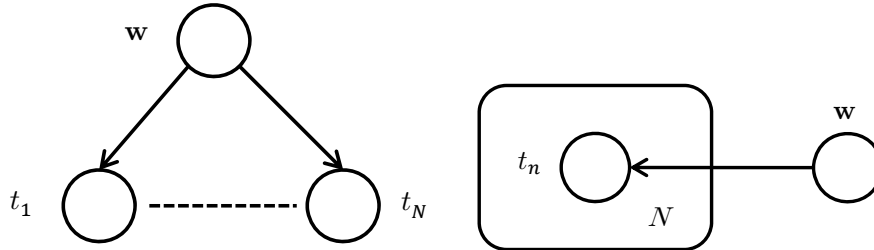
The relationship between directed graph and corresponding distributions over the variables can be generalized easily. For a  $N$  nodes graph the conditional joint distribution for each node conditioned on its parents can be written as

$$p(\mathcal{T}) = \prod_{n=1}^N p(t_n|pa_n) \quad (3.4)$$

where  $\mathcal{T} = \{t_1, \dots, t_N\}$  and  $pa_n$  corresponds to the parents of  $t_n$ . Equation 3.4 reflects the factorization properties of joint distribution of a Bayesian network.

So far we have seen some general examples of how to use directed graphs in order to show probability distributions and how to factorize the conditional distributions based on the graph structure. We now consider a specific example of a polynomial regression model and use a Bayesian network to describe the probability distribution. In this model, polynomial coefficients are represented as vector  $\mathbf{w}$  and observed data  $\mathcal{T} = (t_1, \dots, t_n)$  constitute the random variables of the model. The corresponding directed graph is shown in Figure 3.3(a). It is a bit inconvenient to show multiple nodes of the same type individually in the graph. Therefore, we use plate notation

to express multiple nodes compactly as shown in Figure 3.3(b), where  $N$  indicates the number of nodes of a particular kind.



(a) Directed graphical model representing the joint distribution given in Equation 3.5 corresponding to the Bayesian polynomial regression model. (Source: [5], p. 363)

(b) An alternative, more compact, representation of the graph shown in Figure 3.3(a) in which we have introduced a plate (the box labelled  $N$ ) that represents  $N$  nodes of which only a single example  $t_n$  is shown explicitly. (Source: [5], p. 363)

Figure 3.3: Bayesian polynomial regression model.

From the graph, the joint distribution of  $p(\mathbf{w})$  and  $N$  conditional distributions  $p(t_n|\mathbf{w})$  can be written as

$$p(\mathcal{T}, \mathbf{w}) = p(\mathbf{w}) \prod_{n=1}^N p(t_n|\mathbf{w}) \quad (3.5)$$

We can also use the graph to show the parameters and input data of the model. Small solid circles are used to denote the model parameters when they are deterministic variables and circles when they are random variables. The observed variables are shown by using shaded circles whereas latent variables or unobserved variables are shown by using non-shaded circles. The Figure 3.4 shows the graph of polynomial regression model including the model parameters and corresponding conditional distribution is given in Equation 3.6.

In Figure 3.4 variance  $\sigma^2$  and  $\alpha$  are the parameters of the model with  $\alpha$  as Gaussian prior over  $\mathbf{w}$  and  $\mathcal{X} = (x_1, \dots, x_n)$  represents input data.

$$p(\mathcal{T}, \mathbf{w}|\mathcal{X}, \alpha, \sigma^2) = p(\mathbf{w}|\alpha) \prod_{n=1}^N p(t_n|\mathbf{w}, x_n, \sigma^2) \quad (3.6)$$

Having observed the data  $\{t_n\}$ , we can learn the posterior distribution of polynomial coefficients  $\mathbf{w}$  using Bayes' theorem as given by Equation 3.7

$$p(\mathbf{w}|\mathcal{T}) \propto p(\mathbf{w}) \prod_{n=1}^N p(t_n|\mathbf{w}) \quad (3.7)$$

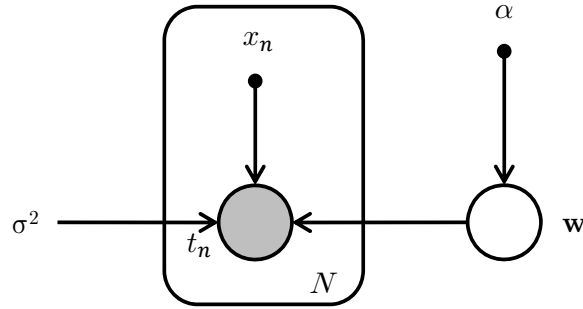


Figure 3.4: Directed graphical model representing the joint distribution corresponding (Equation 3.5) to the Bayesian polynomial regression model with the deterministic variables or model parameters are shown as solid circles. (Source: [5], p. 364)

Our ultimate objective is not to compute the posterior distribution of the model parameters, rather to make prediction for new input values conditioned on the observed data, which actually is an inference problem. If  $\hat{x}$  is the new input value and  $\hat{t}$  is the corresponding distribution conditioned on observed data, then the joint distribution of all the random variables is given by Equation 3.8

$$p(\hat{t}, \mathcal{T}, \mathbf{w} | \hat{x}, \mathcal{X}, \alpha, \sigma^2) = \left[ \prod_{n=1}^N p(t_n | x_n, \mathbf{w}, \sigma^2) \right] p(\mathbf{w} | \alpha) p(\hat{t} | \hat{x}, \mathbf{w}, \sigma^2) \quad (3.8)$$

Using the sum rule of probability distribution we can get the predictive distribution of  $\hat{t}$  by integrating out the model parameters as given in Equation 3.9.

$$p(\hat{t} | \hat{x}, \mathcal{X}, \mathcal{T}, \alpha, \sigma^2) \propto \int p(\hat{t}, \mathcal{T}, \mathbf{w} | \hat{x}, \mathcal{X}, \alpha, \sigma^2) d\mathbf{w} \quad (3.9)$$

### 3.2.1 Generative Models

Directed graphical models or Bayesian networks are often seen as a description of a process by which the data in real world is generated. It shows how the observations can be generated by realization of the random variables while traveling along the edges of the directed graph. As the graphical model captures the *causal* process [94] of observed data generation, therefore, such models are referred to as *generative* models. To explain how the documents are generated in the real world, we specify a language model where each data point corresponds to the words in the document with topics as latent variables. Figure 3.5 represents this model in the form of a graph. In the

graph,  $z$  represents the latent variable (topic) of the model,  $t$  represent the observed variable (terms), while  $\alpha$ ,  $\beta$ ,  $\theta$  and  $\phi$  are model parameters. Given a particular word, our goal is to find a posterior distribution over topics that explains the data best.

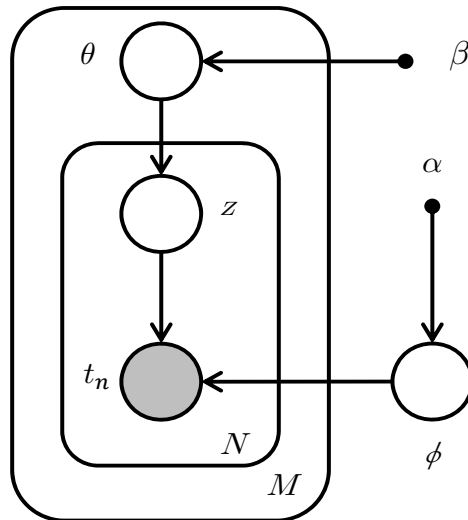


Figure 3.5: A directed graphical model explaining the process by which documents are generated.

The generative process that corresponds to the document generation example is shown in Figure 3.6. The task of the Bayesian inference is to invert generative process as shown in Figure 3.7 and find parameter values that explain the observed data best. From the observed words in a set of documents, we would like to find which language model is most likely to have generated the data. This involves inferring the probability distribution over words associated with each topic, the distribution over topics for each document, and often the topic responsible for generating each word.

### 3.3 Parameter Estimation Techniques (Learning)

In the Section 3.2, we showed how to use directed graphical models to compactly represent probabilistic models. In this section we look at the methods that are common for learning parameters in graphical models. In the parameter estimation, it is assumed that graph structure and dependency relationships between random variables are known. In a Bayesian network the learning task corresponds to finding the parameters  $\theta$  that define the conditional probability distribution of the attributes in a graph with known dependency structure for a given dataset  $\mathcal{X} = \{x_1, \dots, x_N\}$ . Given the observed data and a set of distribution parameter our objective is to find the

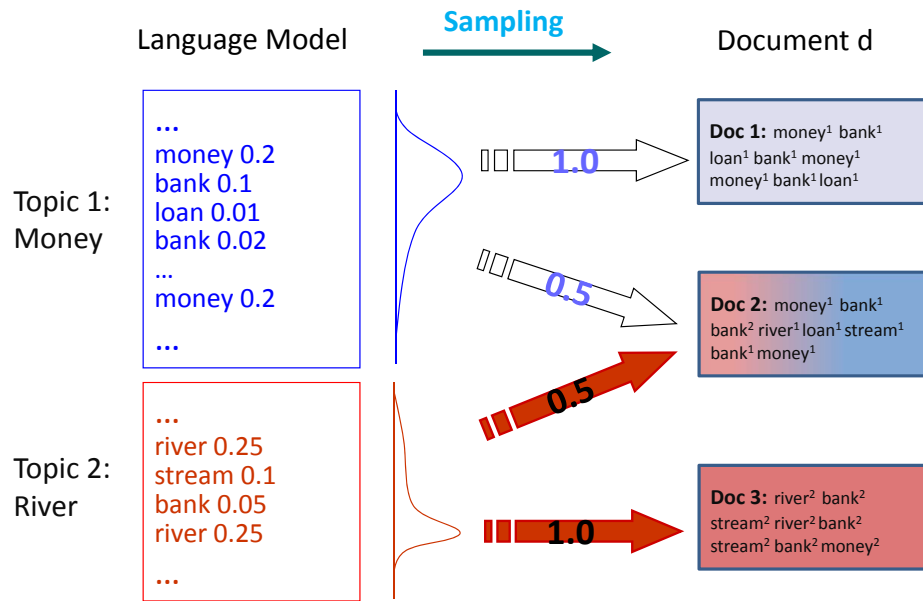


Figure 3.6: A Language model that specifies how the documents are generated in the real world.

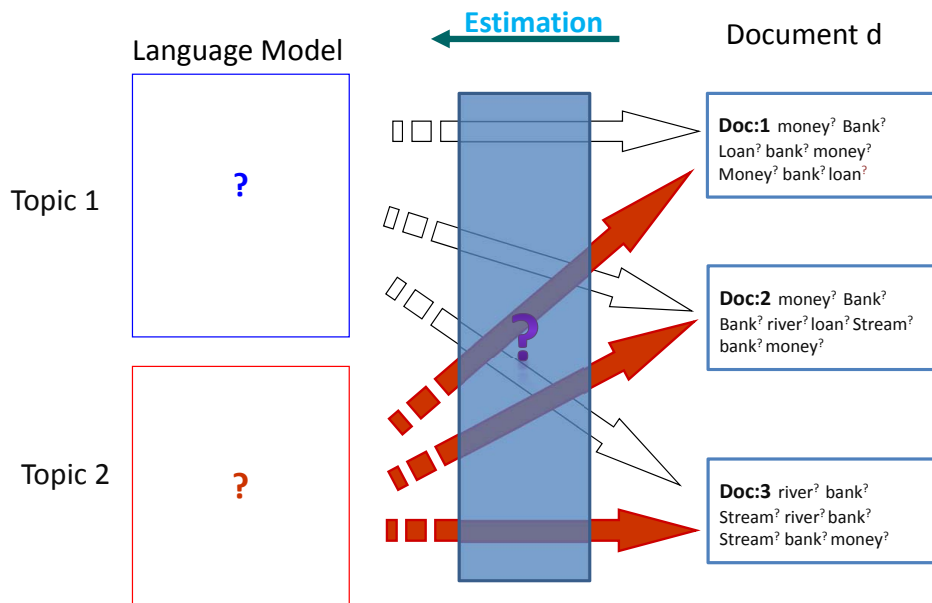


Figure 3.7: Bayesian inference reflecting the process for estimating the parameters of the language model specified in Figure 3.6.



parameters values that are most likely to have produced the data. There exist several approaches such as Maximum likelihood estimation (MLE), Maximum a posteriori estimation (MAP) and Bayesian estimation, which can be employed for parameter estimation. The major component of these approaches is the likelihood function i.e. the probability of the data given the parameters (model) as shown in Equation 3.10.

$$L(\theta|\mathcal{X}) = P(\mathcal{X}|\theta) \quad (3.10)$$

### 3.3.1 Maximum Likelihood Estimation (MLE)

R.A. Fisher developed the principle of maximum likelihood estimation stating that the desired probability distribution is one that makes the observed data “most likely”, and which corresponds to finding parameters that maximize the likelihood  $L(\theta|\mathcal{X}) = P(\mathcal{X}|\theta)$  resulting in parameter vector called MLE estimate. For a given Bayesian network, the likelihood function can be expanded as:

$$L(\theta|\mathcal{X}) = \prod_{n=1}^N p(x_n|\theta) \quad (3.11)$$

Because of the joint distribution of the data  $\mathcal{X}$  Equation 3.11 contains products, therefore, it is convenient to use log likelihood and maximize the log-likelihood function,  $\mathcal{L} \triangleq \log L$ . We can rewrite Equation 3.11 as

$$\hat{\theta}_{ML} = \operatorname{argmax}_{\theta} \mathcal{L}(\theta|\mathcal{X}) = \operatorname{argmax}_{\theta} \sum_{n=1}^N \log p(x_n|\theta) \quad (3.12)$$

In order to obtain parameter estimates, we take partial derivative of the log-likelihood function and solve the system as given in Equation 3.13. By definition, a continuous differentiable function achieves its maximum or minimum on points when its first derivative is zero.

$$\frac{\partial \mathcal{L}(\theta|\mathcal{X})}{\partial \theta_k} = 0 \quad (3.13)$$

However, when a model involves a large number of parameters and its probability density function is highly non-linear then it is usually not possible to obtain an analytic form solution for MLE estimate. In such cases the MLE estimates are sought using non-linear optimization algorithms. The basic idea is to quickly find optimal parameters by searching smaller subsets of the multi-dimensional parameter space instead of exhaustive search over the whole parameter space. Optimization algorithms are prone to *local maxima* because finding optimum parameters is a heuristic process in which the optimization algorithm tries to improve upon initial set of parameters supplied by the user. Depending upon the initial values of parameters, the algorithm could prematurely stop and return a sub-optimal set of parameter

values. There exists no general solution to the local maxima problem but variety of techniques have been developed to overcome this problem though with no guarantee of their effectiveness.

### 3.3.2 Maximum a posteriori Estimation (MAP)

The Maximum a posteriori (MAP) estimation extends the MLE by allowing a prior belief on the parameters. MAP tries to maximize the posterior of the parameters using Bayes' rule.

$$\begin{aligned}
 \hat{\theta}_{MAP} &= \operatorname{argmax}_{\theta} \frac{p(\mathcal{X}|\theta)p(\theta)}{p(\mathcal{X})} \\
 &= \operatorname{argmax}_{\theta} p(\mathcal{X}|\theta)p(\theta) \\
 &= \operatorname{argmax}_{\theta} \{\mathcal{L}(\theta|\mathcal{X}) + \log p(\theta)\} \\
 &= \operatorname{argmax}_{\theta} \left\{ \sum_{n=1}^N \log p(x_n|\theta) + \log p(\theta) \right\}
 \end{aligned} \tag{3.14}$$

Equation 3.14 adds a prior probability  $p(\theta)$  to the likelihood given in Equation 3.12.

### 3.3.3 Bayesian Estimation

In many cases, MLE and MAP estimation over-fits the training data. Bayesian estimation takes the MAP to next level by assuming a distribution over parameters instead of making direct estimates of  $\theta$ . Bayesian method attempts to estimate parameters of an underlying distribution based on the observed distribution. The process begin by assuming a prior distribution which usually is a uniform distribution over the parameters. Given priors, we collect data to obtain observed distribution. Then we compute the likelihood of the observed distribution as a function of parameter values, multiply it with priors and normalize over all possible values to obtain a unit distribution. The resultant distribution is the posterior distribution which is calculated using Bayes' rule.

$$p(\theta|\mathcal{X}) = \frac{p(\mathcal{X}|\theta)p(\theta)}{p(\mathcal{X})} \tag{3.15}$$

In Bayesian estimation, we do not try to find a maximum rather we compute the normalization term  $p(\mathcal{X})$ , whose value can be expressed by the total probability with respect to the parameters.

$$p(\mathcal{X}) = \int_{\theta \in \Theta} p(\mathcal{X}|\theta)p(\theta) d\theta \tag{3.16}$$

Bayesian models often become intricate due to the summations or integrals (c.f. Equation 3.16) of the marginal likelihood which are intractable. Conjugate priors are used to overcome the complexity of Bayesian models. Conjugacy is the property of distribution in which the prior and posterior takes on the same functional form and falls in the same family of distribution but with different parameter values. Dirichlet distribution is an example of conjugate distribution. Posterior distribution in most situations is required for evaluating expectations like making predictions, measuring document similarity or for information retrieval task etc. which is an inference problem.

### 3.4 Inference in Probabilistic Models

Graphical models can be used to answer a variety of queries using posterior distributions. Of these, the most common query type is the *conditional probability query*. In this query type, we compute the probability of new observation  $\bar{x}$  given the data  $\mathcal{X}$ . We generate joint distribution and sum out the joint in case of conditional probability query. This approach to the inference results in exponential increase in the dimensionality of joint distribution making the exact inference intractable. The problem of inference in graphical model is  $\mathcal{NP}$ -hard [62]. The solution in such situation is to use an approximate inference algorithms. There are two classes of approximation schemes according to whether they rely on stochastic or deterministic approximations. Stochastic methods are based on numerical sampling known as Markov chain Monte Carlo techniques [77] such as Gibbs sampling [36], while deterministic techniques are based on analytical approximation to posterior distributions by assuming a particular factorization. The examples of such methods are mean-field variational expectation maximization [6] and expectation propagation [79]. In this section we will provide a brief overview of MCMC technique which generally is used for Bayesian inference.

#### 3.4.1 Markov Chain Monte Carlo Methods (MCMC)

MCMC techniques are often used to solve integration problems in high dimensional spaces. In the context of Bayesian inference, Christophe et al. [3] listed three integration problems which are central to Bayesian statistics and are intractable. For unknown random variables  $x \in \mathcal{X}$  and data  $y \in \mathcal{Y}$ , these integrals are

- *Normalization.* Bayes' theorem requires computing the normalization term to obtain posterior  $p(x|y)$  from prior  $p(x)$  and likelihood  $p(y|x)$ .

$$p(x|y) = \frac{p(y|x)p(x)}{\int_{\mathcal{X}} p(y|\hat{x})p(\hat{x})d\hat{x}} \quad (3.17)$$

- *Marginalisation.* From the joint posterior  $(x, z) \in \mathcal{X} \times \mathcal{Z}$ , we are often required to compute marginal posterior

$$p(x|y) = \int_{\mathcal{Z}} p(x, z|y) dz \quad (3.18)$$

- *Expectation.* Expected value of a continuous variable function is an integral

$$E_{p(x|y)}(f(x)) = \int_{\mathcal{X}} f(x)p(x|y)dx \quad (3.19)$$

Conceptually, integrals in Equations (3.17) to (3.19) requires visiting every element in the space once, measuring the height of the function and adding them all. Instead of visiting each element, one can take a bunch of samples from distribution  $p(x)$  defined on high dimensional space  $\mathcal{X}$  and then take an empirical average over the samples gathered which will give us a good idea of what the integral is. In models involving large number of random variables, sampling directly from posteriors distribution is not feasible. However, we can devise a mechanism which gradually samples from distributions that are closer and closer to the target posterior distribution.

MCMC provides us a framework, which allows sampling from a large class of distributions using Markov chain mechanism. It also scales well with the high dimensional sample space. The Monte Carlo algorithms help to draw  $N = \{x^1, x^2, \dots, x^N\}$  points at random from a target distribution  $p(x)$ . These  $N$  samples are then used to approximate the integrals with tractable sum that converges as follows

$$E_{p(x)}(f(x)) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N f(x^n) = \int_{\mathcal{X}} f(x)p(x|y)dx \quad (3.20)$$

One can draw these  $N$  points using a variety of ways such as adaptive rejection sampling, rejection sampling, Metropolis sampling, importance samples etc. as discussed in Chapter 11 of [5]. A Markov chain [91] mimics a mathematical memoryless system that undergoes transitions from one state to another randomly over a finite number of states so that the next state is only dependent on the current state. In order to construct a Markov chain whose stationary distribution is the posterior of interest, we define a graph and think of these  $N$  points as nodes reflecting the state of the system. We, then randomly traverse the graph moving from one state to another such that the likelihood of visiting any point  $N$  is proportional to  $p(x)$ . We collect independent samples from stationary distribution at each transition in the graph and use them to approximate the posterior. To minimize the influence of initialization parameters, we discard the initial samples and only start recording the samples after a minimum of “burn-in period“ is achieved.

At any given point of time, the target distribution depends on the current state as given below.

$$p_{trans}(x^{n+1}|x^0, x^1, \dots, x^N) = p_{trans}(x^{n+1}|x^n) \quad (3.21)$$

The process is designed in such a way that after enough steps the state of the system reflects the desired stationary posterior distribution. These set of states and transitional model from one state to next forms a Markov chain. Gibbs sampling is a simple and special case of Markov-chain Monte Carlo simulation and is particularly used for inference in high dimensional models such as Latent Dirichlet Allocation.

### 3.4.2 Gibbs Sampling

In Gibbs sampling, the space of Markov Chain is defined over the possible configurations of the hidden variables. The chain runs iteratively sampling at each transition from the conditional distributions of each hidden variable given observations and the current state of the other hidden variables. Gibbs sampling requires  $\mathcal{X}$  to have more than one dimensions such that  $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$  with  $N > 1$ , where each dimension corresponds to a parameter or variable in the model. In Gibbs sampling we do not pick the next state all at once, rather each dimension  $x_n$  in  $K$  is sampled alternatively one at a time conditioned on the values of all other dimensions. The Gibbs sampler adopted from [3, 46, 102] is given in Algorithm 1.

```

Initialize  $x^{(0)} = \langle x_1^{(0)}, \dots, x_N^{(0)} \rangle$ ;
for  $t = 1 \dots T$  do
    • Sample  $x_1^{(t+1)} \sim p(x_1|x_2^{(t)}, x_3^{(t)}, \dots, x_N^{(t)})$ ;
    • Sample  $x_2^{(t+1)} \sim p(x_2|x_1^{(t+1)}, x_3^{(t)}, \dots, x_N^{(t)})$ ;
    ⋮
    • Sample  $x_N^{(t+1)} \sim p(x_N|x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{N-1}^{(t+1)})$ ;
end

```

**Algorithm 1:** Algorithm for Gibbs sampling.

During this process, the new values for the variables are used as soon as they are obtained for calculating the values of remainder variables. This is also evident in the Algorithm 1. For the Gibbs sampler, the full conditionals can be found using Equation 3.22.

$$\begin{aligned}
p(x_n | x_1^{(t+1)}, \dots, x_{n-1}^{(t+1)}, x_{n+1}^{(t)}, \dots, x_N^{(t)}) &= \frac{p(x_1^{(t+1)}, \dots, x_{n-1}^{(t+1)}, x_n^{(t)}, x_{n+1}^{(t)}, \dots, x_N^{(t)})}{p(x_1^{(t+1)}, \dots, x_{n-1}^{(t+1)}, x_{n+1}^{(t)}, \dots, x_N^{(t)})} \\
&= \frac{p(\mathcal{X})}{\int p(\mathcal{X}) d_{x_n}} \quad (3.22)
\end{aligned}$$

If model involves hidden variables  $\mathcal{Z}$  such as Latent Dirichlet Allocation, then we are often interested in posterior distribution of hidden variables given the evidence, which can be computed using Equation eq. (3.23)

$$p(z_k | \mathcal{Z}_{-k}, \mathcal{X}) = \frac{p(\mathcal{Z}, \mathcal{X})}{p(\mathcal{Z}_{-k}, \mathcal{X})} = \frac{p(\mathcal{Z}, \mathcal{X})}{\int_{\mathcal{Z}} p(\mathcal{Z}, \mathcal{X}) d_{z_k}} \quad (3.23)$$

Where integrals in Equations (3.22) and (3.23) change to sum if model involves discrete variables.

## 3.5 Evaluating Topic Models

In the previous sections we looked at different techniques used to learn model parameters and perform inference using the posterior distributions of the parameters. In this section we will look at the applications of these models and some quantitative ways to measure their performance.

Probabilistic topic models are extensively used for text analysis and are applied for tasks such as querying, classification, clustering, prediction and collaborative filtering etc. Latent Dirichlet Allocation (LDA) is one of the simplest example of topic models used for the tasks mentioned above. LDA decomposes a document corpus into a set of latent topics and estimates the associations between latent topic structure and document words. These associations are represented in the form of  $\theta$  (document-topic association and  $\phi$  (topic-word association. The literature [14, 46, 78, 122] reports methods in which we can use and measure the performance of a topic model. Of these, the most common methods include measuring the performance of a topic model on some secondary task such as querying and classification or in predictive performance of the model using held out data.

### 3.5.1 Querying

Querying corresponds to retrieving a set of documents that are relevant for a given query. In querying, we first learn the model from a corpus and for LDA this corresponds to estimating the posterior distribution of  $\theta$  and  $\phi$ . In the next step, for a query document which is unknown to the model previously we estimate its topic structure using already learned model as given in [46] and rank the documents. The ranking is done either by similarity analysis or through predictive likelihood.

**Similarity ranking** involves measuring the similarity between the topic structure of a query document and the topic structure of documents in the corpus. As topic structure implies a distribution, therefore similarity ranking requires measures which can compute the distance between two distributions. The two widely used such measures are Kullback-Liebler divergence [67] and Jensen-Shannon distance [73].

For two probability distributions  $p$  and  $g$  of two discrete random variables  $x$  and  $y$ , the KL-divergence is defined as

$$D_{KL}(p||g) = \sum_i \ln \left( \frac{p(i)}{g(i)} \right) p(i) \quad (3.24)$$

and interpreted as the difference between the cross entropy of  $H(p||g)$  and entropy of  $H(p)$  as given in Equations (3.25) and (3.26) respectively

$$H(p||g) = - \sum_x p(x) \log_2 q(x) \quad (3.25)$$

$$H(p) = - \sum_x p(x) \log_2 p(x) \quad (3.26)$$

which is the information that knowledge of  $g$  add adds to the knowledge of  $p$ .  $D_{KL}$  is non-negative ( $\geq 0$ ), non-symmetric in  $p$  and  $g$  and zero if both distributions are equal.

Originally KL-divergence is non-symmetric, the smoothed and symmetrised Jensen-Shannon extension of KL-divergence is defined as

$$D_{JS}(p||g) = \frac{1}{2} [D_{KL}(p||a) + D_{KL}(g||a)] \quad (3.27)$$

where

$$a = \frac{1}{2} (p + g) \quad (3.28)$$

**Predictive likelihood ranking** involves computing a predictive likelihood that documents in the corpus could be generated by the query and for LDA it is calculated using Equation 3.29 as provided in [46].

$$\begin{aligned} p(\mathcal{W}_d|\tilde{\mathcal{W}}_q) &= \sum_{k=1}^K p(\mathcal{W}_d|z = k)p(z = k|\tilde{\mathcal{W}}_q) \\ &= \sum_{k=1}^K \frac{p(z = k|\mathcal{W}_d)p(\mathcal{W}_d)}{p(z = k)} p(z = k|\tilde{\mathcal{W}}_q) \\ &= \sum_{k=1}^K \theta_{d,k} \frac{n_d}{n_k} \theta_{q,k} \quad (3.29) \end{aligned}$$

Where  $\mathcal{W}_d$  is the word vector for document  $d$  in the corpus,  $\theta_{d,k}$  is the topic distribution of document  $d$  in the corpus,  $\tilde{\mathcal{W}}_q$  is the word vector for query document,  $\theta_{q,k}$  is the topic distribution of the query document,  $n_d$  is document length,  $n_k$  is the number of words associated to topic  $k$  in the whole corpus.

### 3.5.2 Perplexity

Perplexity by convention used in language modeling, is a common measure of performance for unsupervised learning algorithms. We split the dataset into training and test or held-out dataset. The model is learned from the unlabeled dataset and then held-out data is used to measure the generalization performance of the model. Perplexity quantifies the generalization ability of the model to held-out data. It is monotonically decreasing in the likelihood of held-out data and is defined as “the reciprocal geometric mean of the likelihood of a test corpus given the model  $\mathcal{M} = \{\theta, \phi\}$ ”. Formally for a test set  $\tilde{\mathcal{W}}$  of documents it is given in [46] as

$$\begin{aligned} \text{perplexity}(\tilde{\mathcal{W}}|\mathcal{M}) &= \prod_{d=1}^M p(\tilde{\mathcal{W}}_d|\mathcal{M})^{-\frac{1}{N}} \\ &= \exp -\frac{\sum_{d=1}^M \log p(\tilde{\mathcal{W}}_d|\mathcal{M})}{\sum_{d=1}^M N_d} \end{aligned} \quad (3.30)$$

The predictive likelihood of held-out data can be computed by integrating out model parameters from the joint distribution of words in a document. In LDA, likelihood of a document in held-out data given the model is calculated by

$$\begin{aligned} p(\tilde{\mathcal{W}}_d|\mathcal{M}) &= \prod_{n=1}^{N_m} \sum_{k=1}^K p(w_n = t | z_n = k) p(z_n = k | d) \\ &= \prod_{t=1}^V \left[ \sum_{k=1}^K \phi_{k,t} \cdot \theta_{d,k} \right]^{n_d^{(t)}} \end{aligned} \quad (3.31)$$

$$\log p(\tilde{\mathcal{W}}_d|\mathcal{M}) = \sum_{t=1}^V n_d^{(t)} \log \left[ \sum_{k=1}^K \phi_{k,t} \cdot \theta_{d,k} \right] \quad (3.32)$$

where  $n_d^{(t)}$  is the count of term  $t$  in document  $d$  and  $\theta_d$  is the topic distribution of the document  $d$  and  $\phi_{k,t}$  is the term distribution of topic  $k$  in held-out dataset. When comparing models, lower perplexity scores for a model indicate that it generalizes better to unseen documents.



## 3.6 Summary

In this chapter, we presented a generalized overview of probabilistic topic modeling and described the representation, learning and inference techniques used for graphical models. We introduced methods for estimating model parameters and ways to perform inference using approximation algorithms. We also touched the topic of evaluation in topic models and listed some common methods which are used for measuring model performance in the context of Latent Dirichlet Allocation. Further discussions, explanations and detailed mathematical formulations of the techniques discussed in this chapter can be found in [5, 37, 46, 63, 102].



## Chapter 4

# Searching Microblogs: Coping with Document Quality and Sparsity

Content publishing and sharing has been made easy by online information sharing platforms also referred to as social networking platforms, where we are no longer bound to share contents using traditional print media. These social media contents produced online are unedited raw content and bear no influence of corporate-owned big media print organizations, and thus are a valuable source of first hand real life information. The examples of such social platforms include microblogging platforms such as Twitter and social networking platforms like Facebook, Myspace, Google+ etc. In these platforms people can subscribe to other people whom they trust for receiving shared information.

However this liberty of information sharing does have some cost associated with it when you compare it with print media. The contents in the print media are carefully crafted, edited, length pruned, well structured, without grammatical errors, confined to well defined language and present quality information which is interesting and useful to a wider group of people. Contrary to this, information shared via online social platforms lacks the control of contents, vary in quality and are generally sparse but nonetheless still are valued as rich source of information. Mining quality information which is interesting to general audience from unstructured, sparse contents poses certain challenges in determining content quality and retrieving interesting and useful information. In this chapter, we look at the problem of retrieving and measuring content quality in microblogging portals Twitter<sup>1</sup>.

The rest of this chapter is structured as follow: In Section 4.1, we introduce microblogging environment and an overview of the retrieval problems with the list of our contributions in these settings, while Section 4.2 lists the

---

<sup>1</sup><http://twitter.com/>

datasets used for experiments in this chapter. In Section 4.3, we analyze microblog contents and train a regression model to find out which content features are important for a tweet to be retweeted and thus contribute towards the interestingness of a microblog. Based on the weights learned for different features, we propose a static content quality measure of microblogs. Section 4.4 describes the details of retrieval problems in microblogs and suggests the use of content quality measure proposed in Section 4.3 for solving these problem. In Sections 4.6 and 4.7, we provide evaluation results of our approach of measuring microblogs content quality and its effectiveness in retrieval scenarios. Section 4.8 covers the existing research in this area and Section 4.9 summarizes this chapter.

## 4.1 The Microblog Environment

A microblogging platform such as Twitter allows the users to share information via short messages referred to as tweets. A tweet is distributed to those people that subscribe to the information of the author. In the context of Twitter this subscription is known as *following*. This structure of followers forms a large network among the users of Twitter. A particularity is that the receiver of a message has the option to relay it and forward it to her followers. This practice is called *retweeting* and is used by some users as a measure of content quality for tweets [7]. By convention, retweets are indicated by specific keywords such as RT, via or retweet button.

The question of what causes a message to be retweeted has frequently been addressed, but mainly in a scenario of retweet prediction for a given user and with a focus on the structure of the social network [13, 69, 104]. Studies of tweets and retweets have revealed that the context of a tweet influences its actual chance to be retweeted [117]. Most prominent context features are the social graph and the time of the original tweet. In this case, a typical observation is that a well connected user with active followers is more likely to be retweeted. As the content of a tweet in such a setting is neglected or reduced to a few very simple features, a network-based analysis of retweets may give hints into *who* tends to write interesting messages, but cannot give insights into *what* the community is interested in. As our focus is on the message itself, we will deliberately ignore such context information, and rely only on features extracted from the message itself.

In Twitter, the length of messages is restricted to 140 characters. While the conciseness of messages has been cited as a major reason for Twitter's success, it is at the same time problematic for text retrieval: Term frequencies, for instance, are typically used as a parameter in the estimation of term importance within a given document. In short texts, however, this essential feature does not discriminate much between different documents, as it is nearly a binary value. In fact, our analysis across several large datasets

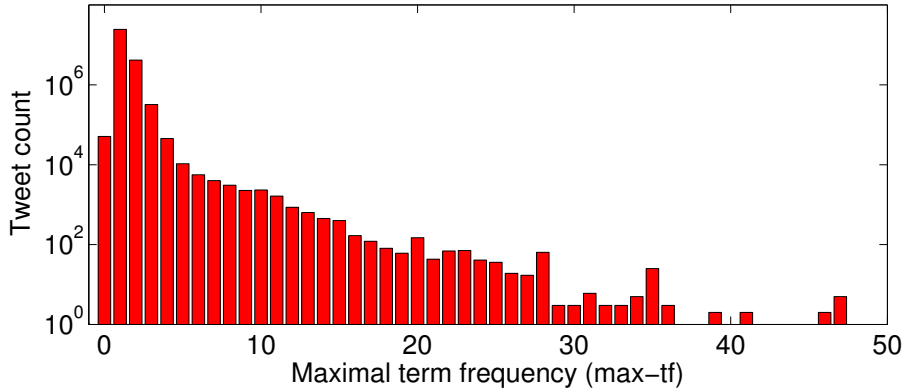


Figure 4.1: Distribution of maximal term frequencies ( $max\text{-}tf$ ) in Twitter messages of the CHOUDHURY-EXT dataset after removing stop words.

shows that about 85% of all Twitter messages contain each term at most once (see Figure 4.1). In consequence, most retrieval models are effectively reduced to using global term weights, that measure the discriminativeness of terms.

A second challenge for retrieval on microblogs is the nature of the medium. With regard to the purpose, Twitter documents are different from documents in the classical sense. Classical documents address a wider, open audience, as in principle everyone has access to the document. Microblog messages, instead, typically have a more restricted and better defined audience provided by the social network of a user. Thus, while the primary purpose of a tweet can be to communicate information (e.g. news, sharing resources, broadcast an alert) it might, alternatively, also serve other primary purposes, such as social interaction, promotion, requesting feedback or expressing emotions [18]. In a general retrieval scenario, the rather private and personal messages in a social interaction context are typically of less interest for a user with a concrete information need. This implies that a retrieval criterion for microblog messages should relate to the *interestingness* of a tweet.

Information retrieval on microblogs needs to address these two challenges that are immanent to the technical and social context of the medium. Classical retrieval models do not consider these aspects; they are directed to longer texts and assume the intention of a document to primarily be the transmission of information. Also, recent approaches transferring the concepts of authority in reference networks, such as citations or hyperlinks, to the social network of Twitter do not take into consideration that the semantic of the social network is not equivalent to a content motivated reference network.

One possibility to overcome such a problem is to introduce static quality

measures. Static quality measures such as Google’s PageRank [9] capture the quality of an information item independently of any query. They can be used to improve ranking once the presence of a keyword is observed. Hence, the focus of this work is to elaborate a content-based quality measure for short texts.

As a static quality measure for Twitter, we propose the *retweet* function. As a retweet is relayed to all the followers of a user, the message being retweeted can therefore be considered of general interest. This notion of a tweet being potentially interesting to other users is then a suitable way to capture static content quality independently of a query. Thus, we consider a tweet to be of quality if it is interesting for readers.

The contributions of our work in this chapter are on the following levels:

- We consider the problem of learning which tweets are retweeted based on a wide range of content features and independently of context information such as the user’s position in the social network and the timestamp of a tweet. We show that it is possible to predict which tweets are retweeted.
- By analyzing the parameters learned in our prediction model, we identify the features that contribute most strongly to the probability of a tweet being retweeted. This allows for a deeper insight into what is of interest in the Twitter community.
- We analyze term and length features of microblog messages and provide theoretical and empirical evidence that length normalization introduces an unjustified bias for Twitter.
- We introduce a static quality measure “*interestingness*” and show that it improves retrieval results in particular on short, underspecified queries.

## 4.2 Retweet Datasets

In our experiments for learning the feature weights and evaluating the use of “interestingness” as static quality measure in ranking of the microblogs in a retrieval task, we use three established Twitter datasets. They are listed in Table 4.1 with some key properties and statistics. All datasets consist of a corpus of individual tweets, along with their timestamp and an identification of the user who sent the tweet.

We detect retweets by using the patterns given in Table 4.2 that capture the different ways people mark retweets. The patterns are applied in a case-insensitive way.

Table 4.1: List of established Twitter datasets used in our experiments.

Dataset	Users	Tweets	Retweets
CHOUDHURY [21]	118,506	9,998,756	7.89%
CHOUDHURY-EXT [21]	277,666	29,000,000	8.64%
PETROVIĆ [96]	4,050,944	21,477,484	8.46%

Table 4.2: List of patterns used by the people on Twitter to mark retweets.

RT @[username] ...
... (via @[username])
retweeting @[username] ...
🔄 @[username] ...
retweet @[username] ...

### 4.3 Content-based Retweet Prediction

As mentioned in Section 4.1, we focus on the content of a tweet and train a prediction model to forecast for a given tweet its likelihood of being retweeted based purely on its contents. From the parameters learned by the model we deduce what are the influential content features that contribute to the likelihood of a retweet – and thereby are characteristics of an interesting message in the context of Twitter.

For this purpose, we analyze a set of high- and low-level content-based features. The low-level features comprise the words contained in a tweet, the tweet being a direct message, the presence of URLs, hashtags, usernames, emoticons, and of question and exclamation marks as well as terms with a strong positive or negative connotation. These features are directly extracted from the text of a message and do not require further processing. The high-level features are formed by associating tweets to topics and by determining the sentiments of the tweets. For retweet prediction, we employ a logistic regression analysis model.

We are interested in retweets, because they can be seen as an indicator for interestingness. The rationale behind this hypothesis is that the user retweets a message when she considers the original tweet interesting enough to relay it to her own followers. However, whether a particular tweet actually is retweeted depends heavily on context, such as the user’s position in the social graph or the time of day the tweet is posted. Generally, a tweet of a user with few or passive followers is less likely to be retweeted. Similarly, tweets posted in the night tend to get retweeted less. Despite this, neither of these contextual pieces of data has any influence on the content of a tweet. To avoid introducing such a contextual bias into our analysis of interestingness, we deliberately ignore such context information and rely only

Table 4.3: The features and their value range used to represent tweets.

Feature	Values
Direct message	{0, 1}
Includes username	{0, 1}
Includes hashtag	{0, 1}
Includes URL	{0, 1}
Exclamation mark	{0, 1}
Question mark	{0, 1}
Term positive	{0, 1}
Term negative	{0, 1}
Emoticon positive	{0, 1}
Emoticon negative	{0, 1}
Valence	[-5, +5]
Arousal	[-5, +5]
Dominance	[-5, +5]
Terms	[0, 1]
Topics (100)	[0, 1]

on features extracted from the message itself. We proceed with a detailed description of the features we actually use for the representation of tweets.

### 4.3.1 Features

All of the following features are based on the tweets themselves and ignore a tweet’s author and timestamp. A complete list of the employed content features is given in Table 4.3. As can be seen there, most features are binary, i.e. have a value of either 0 or 1.

**Direct messages.** Direct messages are addressed to another user directly.

These messages start with the username of the addressee. While other users can still see these messages<sup>2</sup>, they are not in the primary focus of the message. Direct messages are meant as kind of public conversation, rather than a general broadcast of information.

Given the rather personal note and intention of direct messages, as well as their different purpose in the interaction among users, we expect them to be much less retweeted. Accordingly, the feature of whether a tweet is a direct message is of importance for our retweet prediction.

**URLs, usernames and hashtags.** Without further differentiation we consider the presence of particular items typical for tweets. These are the

<sup>2</sup>Unlike private messages which are visible only to the sender and recipient.



presence of a URL, the mention of a username or a hashtag. Usernames are used in Twitter to refer to other users directly, either for addressing a user or for talking about him. Hashtags, or simply tags, are used to mark specific topics. They can be either inline in the messages or appended after the message itself. URLs are universally used to indicate the location of the full text being talked about. On Twitter, usernames and hashtags can be identified by their specific syntax using the pattern `@username` and `#hashtag`. We use the string `http:` to identify URLs. These give three binary features.

Related work has already recognized the effect of the presence of URLs, hashtags and usernames on the retweet behavior.

**Exclamation and question marks.** We use the presence of exclamation marks “!” and question marks “?” at the end of tweets as two binary features. Exclamation marks are used in personal communication to mark strong and potentially emotional statements and in general text to mark interjections and exclamations. Question marks indicate questions in all types of text, and are by their nature intended to elicit responses. Due to the multiple uses of both symbols, we cannot easily judge if in all cases a question mark really does indicate a question or an exclamation mark expresses a strong statement. However, using the location at the end of the message as an indicator is a suitable and straightforward heuristic.

Both types of messages might have an influence on the reaction of the users that receive such a tweet. Questions can be passed on in order to extend their reach and find an expert capable of providing an answer. Statements might be retweeted to demonstrate support.

**Positive and negative terms.** We look for positive and negative words from the short predefined list given in Table 4.4. Terms expressing positive and negative feelings have previously been found to influence the social interaction in Twitter [95], and we conjecture them to also play a role in making a tweet interesting or uninteresting.

Following the line of thought on statements marked with an exclamation mark, strong positive and negative terms might foster a retweet as a sign of support among users.

**Emoticons.** Emoticons or smileys are short character sequences representing emotions. We parse the tweets to find positive emoticons such as :-)) and negative emoticons such as :-(, giving two binary features. Table 4.4 gives the complete list.

As emotions have been observed to influence reaction among users, emoticons might be an indicator of interestingness. Besides transmitting emotions, they are also used to mark jokes, funny comments or

Table 4.4: Terms and emoticons expressing positive and negative emotions in Twitter messages.

	Positive	Negative
<b>Terms</b>	great like excellent rock on	fuck suck fail eww
<b>Emoticons</b>	:-) :) ;-)	:( :(

irony. These kind of messages have a tendency to be passed on, as can be observed by the behavior of people forwarding emails of that kind.

**Sentiments.** Many tweets are personal and express sentiments. To detect the sentiments expressed by a tweet, we follow previous Twitter research and select a simple dictionary-based approach [60]. We use the Affective Norms of English Words (ANEW) dictionary [8], which gives for 1,030 English words numerical values that capture valence (pleasure vs displeasure), arousal (excitement vs calmness) and dominance (weakness vs strength of expressiveness).

In order to deal with inflections of dictionary words, we apply the Porter stemmer [99] to both the dictionary terms and the words extracted from the tweets. The computed values vary from 1 to 10, and we normalize them by subtracting the median value 5. This allows positively and negatively annotated terms to counterbalance each other. The total valence, arousal and dominance of a tweet are computed as the sum of the values associated with each term. Words not contained in the ANEW dictionary are considered neutral and do not affect the score for these features.

The three dimensions we used in this setting capture different notions of sentiments. This allows for a more subtle analysis than the more common sentiment analysis techniques focusing on positive and negative emotions.

**Terms.** The most obvious content feature in text are the contained terms. We extract terms and normalize them using case folding and the Porter stemmer [99]. Given the sparsity of tweets and the reduced expressiveness of the frequency of a term in a message we only consider presence or absence of each individual term and ignore multiple occurrences. For each message  $M$  we compute the odds of it being a retweet based on the terms  $t_i$  it contains. Assuming independence between the occurrences of terms and employing Bayes' theorem the odds value can

be brought into a form that is easier to handle:

$$\begin{aligned}
 O(\text{retweet} \mid M) &= \frac{p(\text{retweet} \mid M)}{p(\text{non-retweet} \mid M)} \\
 &= \frac{p(\text{retweet}) \cdot p(M \mid \text{retweet})}{p(\text{non-retweet}) \cdot P(M \mid \text{non-retweet})} \\
 &= O(\text{retweet}) \cdot \frac{p(t_1 \dots t_n \mid \text{retweet})}{p(t_1 \dots t_n \mid \text{non-retweet})} \\
 &= O(\text{retweet}) \cdot \prod_{t \in M} \frac{p(t \mid \text{retweet})}{p(t \mid \text{non-retweet})}
 \end{aligned}$$

where  $O(\text{retweet})$  are the a priori odds of a retweet, and the product ranges over the ratios of the probabilities of each contained term to occur in a retweeted or a non-retweeted message. To estimate these probabilities we use maximum likelihood estimation and Laplacian smoothing [70] to handle unseen terms.

Even though the sparsity of tweets makes it difficult to train a prediction model on terms alone, the individual terms are a very good representation of the content. Thus, the contained terms can be seen as a very detailed and narrow description of the tweet’s latent topic. The topic models described below provide a broader approach for capturing the topic orientation of the tweet content.

**Topics.** The topic of a tweet is a latent feature and can be inferred by analyzing a tweet’s content. As each tweet is limited to 140 characters with heterogeneous vocabulary written in a language unlike standard written English, many supervised models in machine learning and natural language processing are hard to train and evaluate on tweets. Modeling Twitter content requires methods that are suitable for short texts with heterogeneous vocabulary with minimum supervision. Recent work shows that one such method which works well on short texts for modeling topics is Latent Dirichlet allocation (LDA) and its extensions [6, 126]. In LDA a topic is represented as a distribution over words that occur typically for this topic.

To learn latent topics from training and test data we construct a topic model using Gibbs sampling for latent Dirichlet allocation. We use 100 latent topics for our datasets. The number of topics for the corpus is an objective criterion that can be chosen using a number of available methods. A solution with too few topics will generally result in broad topics whereas a solution with too many topics will result in fine grained topics that are hard to interpret. Our approach is to use perplexity (as explained in 3.5.2) to choose the number of topics that leads to the best generalization performance for the task. The

perplexity of a model describes its entropy and has been used to assess generalizability of text models to subsets of documents [6].

Topic features are broader in concept than individual words, since a single topic consists of an entire collection of related words. Thus, the LDA topics can be used to understand which larger topics are influential on the retweeting behavior of users.

### 4.3.2 Regression Analysis

We use logistic regression to compute the probability of a new tweet being retweeted. Logistic regression is a generalized linear regression method for learning a mapping from any number of numeric variables to a binary or probabilistic variable [50]. In the Twitter setting, we learn a mapping from the features of a tweet to the binary value indicating retweets.

Let  $f_{ij}$  be the feature  $i$  of tweet  $j$ , and  $\text{retweet}_j$  the 0/1 variable indicating whether the tweet  $j$  was retweeted. Logistic regression learns weights  $w_i$  under the following model:

$$p(\text{retweet}_j | f) = \frac{1}{1 + e^{-(w_0 + \sum_i w_i f_{ij})}} \quad (4.1)$$

The weights  $w_i$  learned by logistic regression can be interpreted as the log-odds for the feature  $i$ . Therefore, positive weights denote a higher probability of retweet for tweets having this feature of  $p > 1/2$ .

Once we have trained the logistic regression model we obtain feature weights that indicate their influence on the probability of a message being retweeted. By looking at these weights, we can understand what influences the retweet behavior in Twitter and in conclusion can deduce assumptions on what the users consider interesting on a global scale.

By calculating the features for a new message and applying the function defined in Equation (4.1) we obtain a probability for this new message to be retweeted. The computed probabilities can be used for two applications: as a measure for predicting whether a tweet will be retweeted, and as a measure for interestingness.

### 4.3.3 Accuracy of Retweet Prediction

In order to verify the learned model parameters, we measure the accuracy of retweet prediction. Therefore, we split the set of tweets into a training and a test set based on the timestamps of the tweets. The training set consists of all tweets with the lowest timestamp values and contains 75% of the available dataset. The remaining 25% of the data are retained for the test set on which we evaluate the prediction quality. In PETROVIĆ dataset,

the training set contains 7.78% retweets while test set consists of 10.49% retweets.

As described in Section 4.3.2, we then compute all features for the tweets in the training and test sets. For features that require a model such as word odds and topics, we compute this model only for the training set. Logistic regression is then applied to the features in the training set. The resulting weights are finally used to compute the probability of tweets in the test set to be retweeted.

Figures 4.2 to 4.4 show the accuracy of retweet prediction in form of a ROC (receiver operating characteristic) curve. A ROC curve is a method to visualize the prediction accuracy of ranking functions showing the number of true positives in the results plotted against the number of results returned. A ROC curve generated by a random rank would result in a straight diagonal line and rankings performing better than a random rank result in a line going over that diagonal. Figure 4.2 shows the ROC curve for prediction by logistic regression on the PETROVIĆ dataset. The plot also contains a separate curve for each feature used separately. For features  $i$  that have a negative weight  $w_i$  learned by logistic regression as shown in Table 4.5, we show the ROC curve of the inverse ranking.

As expected, prediction is most accurate when taking into account all features. Individual features that perform well for retweet prediction are term odds and the detection of direct messages. Similar results are obtained for CHOUDHURY dataset. For CHOUDHURY-EXT dataset, term odds performs poor than the random prediction. We interpret this as terms playing a role in distinguishing types of tweets such as news, personal messages, etc. We conclude that only certain types of messages are likely to be retweeted.

#### 4.3.4 Analysis of the Weights

Now that we have verified that our model does not make random predictions, but does capture the probability of a tweet being retweeted, we can analyze the weights we have obtained for our model. Table 4.5 lists the weights learned using logistic regression for different features for the CHOUDHURY-EXT dataset. The weight  $w_i$  of a binary feature  $i$  with possible values 0/1 learned by logistic regression can be interpreted as the log-odds of a tweet having that feature:

$$w_i = \ln \left[ \frac{p(\text{retweet}_j \mid f_{ij} = 1)}{p(\text{retweet}_j \mid f_{ij} = 0)} \right]$$

From the learned regression weights for features, we can make some interesting observations:

- Direct messages are unlikely to be retweeted, which is indicated by the strong negative weight associated to the according feature. This

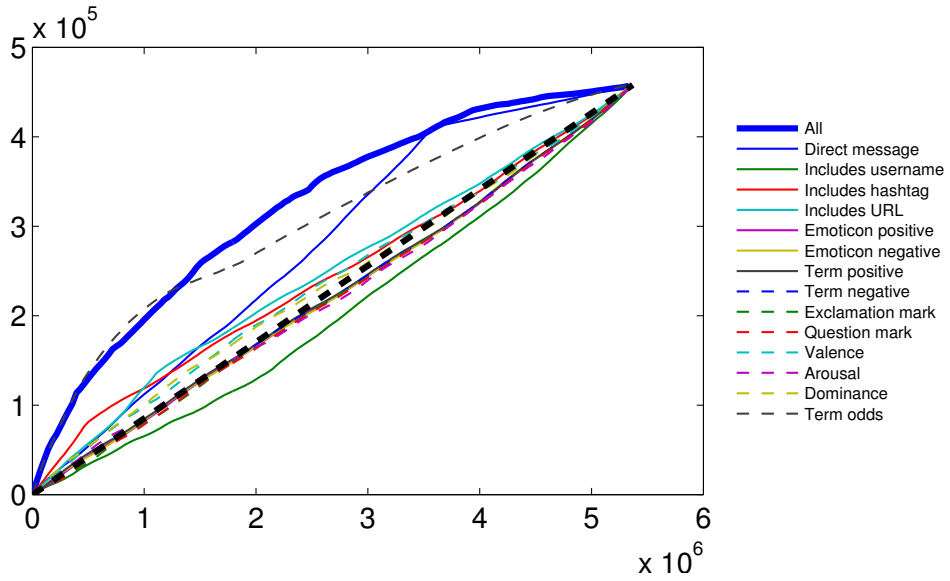


Figure 4.2: The accuracy of retweet prediction using logistic regression based on all features, and of each feature separately, in the PETROVIĆ dataset. The accuracy is represented as a ROC curve. For clarity, ROC curves for topics are not shown.

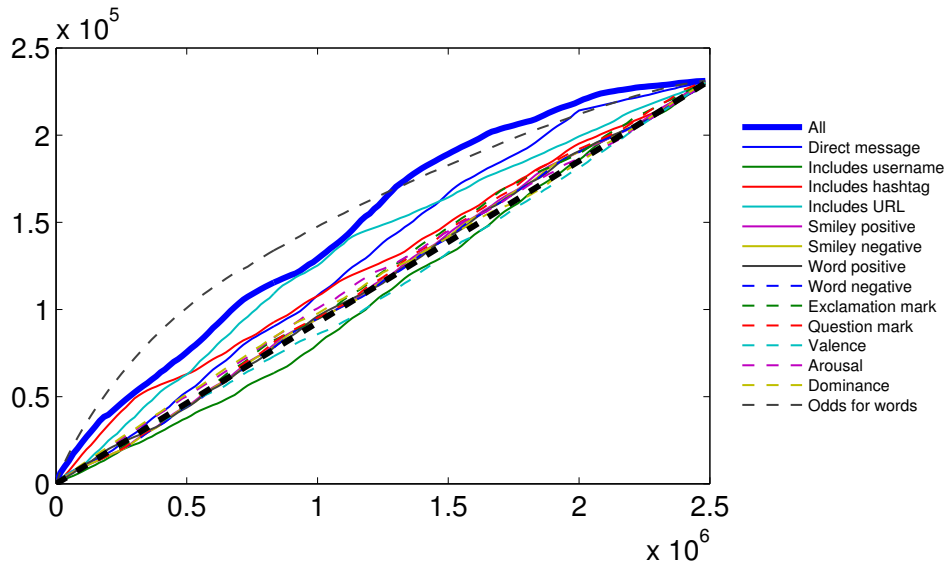


Figure 4.3: The accuracy of retweet prediction using logistic regression based on all features, and of each feature separately, in the CHOUDHURY dataset. The accuracy is represented as a ROC curve. For clarity, ROC curves for topics are not shown.

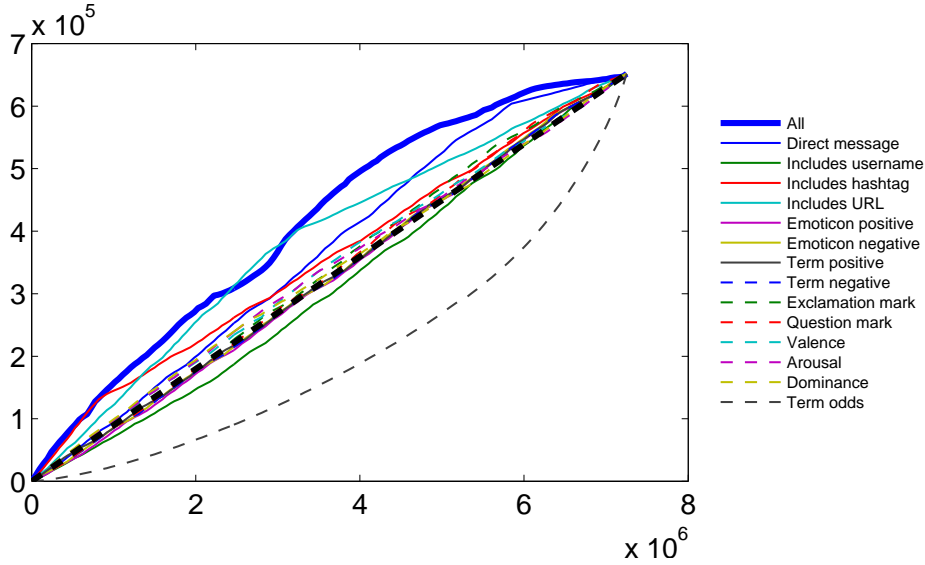


Figure 4.4: The accuracy of retweet prediction using logistic regression based on all features, and of each feature separately, in the CHOUDHURY-EXT dataset. The accuracy is represented as a ROC curve. For clarity, ROC curves for topics are not shown.

Table 4.5: Weights of features learned by logistic regression on the CHOUDHURY-EXT dataset. Positive values denote a positive contribution to a tweet being retweeted; negative weights denote a negative contribution to a tweet being retweeted.

Weight $w_i$	Feature $i$
-147.89	Direct message
146.82	Includes username
42.27	Includes hashtag
249.09	Includes URL
-16.85	Exclamation mark
23.67	Question mark
13.66	Term positive
8.72	Term negative
-21.80	Emoticon positive
9.94	Emoticon negative
-26.88	Valence
33.97	Arousal
19.56	Dominance
19.79	Term odds

observation corresponds to our intuition, that personal messages are not of interest on a global scale.

- Messages with hashtags, usernames and URLs are likely to be retweeted. This observation has already been made in related approaches which considered these features alone. However, looking at the prediction performance of these features individually as shown in Figure 4.2, we can see that they cannot be applied in isolation but that for predicting retweets they need to be combined with other features.
- We also observe sentiments to play an important role for retweeting. Note that the weights need to be interpreted in a slightly different manner in this case. As the features can have negative and positive values (corresponding to the two poles for each sentiment feature), a negative weight does not imply a negative impact on the probability for a retweet. Rather, a negative weight is a sign that negative values for this feature increase the probability for a retweet, while positive weights indicate a better chance for a message to be retweeted if also the feature shows a positive value. Thus, tweets with negative valence values, i.e. annoying or displeasing contents, tend to get retweeted more often. Likewise tweets with positive arousal and dominance values, i.e. exciting and intense tweets, are more likely to be retweeted. This seems to confirm the idiom that bad news travels fast.
- Also, including a positive emoticon such as :-)) lowers the probability of retweet, whereas adding a negative one such as :-( increases the probability. By relating the negative emoticons to negative and displeasing emotions, this seems to support the observations made above for sentiments.
- Positive and negative terms from our short list in Table 4.4 both render a tweet more likely to be retweeted. In this case, positive words have a stronger effect. One possible explanation is, that users are slightly more reluctant to retweet messages containing rude terms. In any case, these extreme and strong words seem to stimulate a reaction in the followers.
- Tweets ending in an exclamation mark are not likely to be retweeted, but tweets ending in a question mark are. This is an interesting observation and would motivate a deeper analysis of the social aspects on Twitter in question answering, i.e. if questions are really passed on to find an expert capable of answering them.
- Terms are a strong indicator for a retweet. As already seen in the evaluation of the prediction quality, the content has a strong influence on the probability of a message to be retweeted.



Table 4.6: Logistic regression weights and corresponding high probability terms that describe a particular topic in the CHOUDHURY dataset. The weights can be interpreted as the log-odds of a tweet from a given topic to be retweeted. Positive weights indicate topics that are likely to be retweeted and negative weights indicate topics that are unlikely to be retweeted.

Weight $w_i$	Topic $i$
27.54	social media market post site web tool traffic network
16.08	follow thank twitter welcome hello check nice cool people
15.25	credit money market business rate economy home
2.87	christmas shop tree xmas present today wrap finish
-14.43	home work hour long wait airport week flight head
-14.43	twitter update facebook account page set squidoo check
-26.56	cold snow warm today degree weather winter morning
-75.19	night sleep work morning time bed feel tired home

The topic features are not included in the previous list because they need to be discussed in a more differentiated way. As there are 100 different topics, we cannot address all of them individually. Rather we report the trends we have observed with respect to the topic features.

Table 4.6 shows the four topics having highest log-odds with positive weights (top high-probability terms for each topic) based on the logistic regression score of the training data that are most likely to be retweeted, and four topics having lowest log-odds with negative weights that are least likely to be retweeted based on regression analysis of training data. From the analysis results it is clear that topics that are very likely to be retweeted address broader public interests such as social media and social networking in general, economy and Christmas-like holidays and public events. Topics that are least likely to be retweeted based on regression scores are more specific and individual in nature, reflecting personal tasks, moods and observations.

### 4.3.5 Example: Interesting Tweets

Given the notion of interestingness we can obtain from the odds for a tweet to be retweeted and allows for realizing practical applications as shown in Section 4.4. For instance, it is possible to get the most interesting tweets from a dataset about a specific topic. As an example, we have listed the top ten most interesting tweets with respect to the log-odds of predicted retweet probability for the term *Recipe* in Table 4.7.

Table 4.7: Top 10 interesting tweets by the log-odds of predicted retweet probability for the query *Recipe* in the CHOU DHURY dataset.

Log-odds	Tweet
3245.00	How to make potato latkes video recipe by @hand-madekitchen <a href="http://tinyurl.com/n22t4p">http://tinyurl.com/n22t4p</a> #cooking #recipe
2455.30	Recipe for Chinese Chicken Congee inspired by a painting from the Sung Dynasty <a href="http://bit.ly/16V5L0">http://bit.ly/16V5L0</a> #art #food #foodie #recipe
2439.56	Have a great idea for a recipe using @greensbury organic meats? You could win free #meat and get your recipe posted!
2385.60	New Raw Food World S Raw Ice Cream Recipe, Episode #134: We've got a Raw Ice Cream Recipe JU.. <a href="http://tinyurl.com/pdt7cq">http://tinyurl.com/pdt7cq</a>
2362.94	Recipe looks good - Potatoes Gribiche Recipe: I've not really been in the mood for winte.. <a href="http://tinyurl.com/cay294">http://tinyurl.com/cay294</a>
2337.91	what to pack for a day at the beach with the fam (plus a yummy beach pasta salad recipe) <a href="http://is.gd/1sBKM">http://is.gd/1sBKM</a> #ocmom #recipe
2301.83	Tasty pasta cake recipe's :- ) Bub Hub Pregnancy & Parenting Forum: Tasty pasta cake recipes Recipe.. <a href="http://bit.ly/ONk9l">http://bit.ly/ONk9l</a>
2294.25	It's Taco Tuesday! How about making some Buffalo Sausage Tacos at home, great recipe: <a href="http://bit.ly/jwPDT">http://bit.ly/jwPDT</a> yummm! #food #recipe
2285.98	Great grilling recipe for this weekend: Cranberry-Onion Pork Roast, Check out the recipe in the Hotlanta Forum: <a href="http://tr.im/s8HA">http://tr.im/s8HA</a> #food
2200.94	RT @nytimesdining: NYT Recipe Challenge #nytrc: Tweet this recipe in as few characters as possible. Serial tweets ok. <a href="http://bit.ly/bhf92">http://bit.ly/bhf92</a>

#### 4.4 Retrieval on Microblogs

In this section, we consider the two challenges for retrieval on microblogs which we have already mentioned in the Section 4.1: sparsity and quality. We look at the impact of sparsity on length normalization in retrieval models and motivate to ignore document length in a microblog scenario. Further we introduce a way to incorporate the notion of “interestingness” and retweet odds discussed in Section 4.3 as static measure of content quality for mea-

asuring quality in tweets and show that it helps to overcome the problem of underspecified queries and can compensate for the lack of meaningful term frequencies in microblog messages.

#### 4.4.1 Term Sparsity and Length Normalization

We already mentioned that microblog messages contain few terms in general and very rarely contain a term more than once (c.f. Figure 4.1). This observation can clearly be attributed to the intrinsic length restrictions of microblog messages. Intuition dictates that this term sparsity will have an impact in a retrieval setting. The most obvious impact is that a potentially relevant tweet will not be retrieved at all if among its few terms it does not contain one of the query terms. This risk is much higher than with classical documents, as the length restrictions prevent an author from using synonyms or elaborating concepts with additional words. But, a second and more subtle impact lies in the length restriction itself, as we will see in a moment.

Length normalization is an essential ingredient to modern retrieval models. The motivation for length normalization is to counterbalance the potential advantages of longer documents [103, 114] that are commonly explained based on the *verbosity hypothesis* and the *scope hypothesis*.

**Verbosity Hypothesis:** A long document elaborates the same topic longer and repeatedly. Therefore it also contains the same terms repeatedly while not adding further information to the document. This leads to a higher term frequency and in consequence to higher weights for the repeated terms. The best – though artificial – example for verbosity would be a concatenation of twice the same document. Obviously, such a long document should not be preferred over a short document, which essentially contains the same information.

**Scope Hypothesis:** A long document addresses several topics. Therefore it contains more different terms and might seem relevant to wider range of different queries. The general line of thought here is that a user would prefer a short and focused document over a long document relating to several topics.

Given the restriction of microblog messages to contain very few if not even a limited number of terms, intuition tells us that neither the verbosity nor the scope hypothesis can serve to explain the document length in Twitter messages. To verify our intuition, we analyze a large collection of Twitter messages with respect to two questions:

- can we observe a tendency of longer tweets to be verbose, i.e. contain terms repeatedly and

- can we observe a tendency of longer tweets to have a larger scope, i.e. to cover several topics.

To discover verbosity we look at redundancy in a tweet, i.e. how many terms appear more than once and how this correlates with document length. As a measure for document length, we employ once the number of characters and once the total number of terms. Given the non-normal distribution of document length (c.f. Fig. 4.5 for the distribution of messages w.r.t. character length) we calculated Spearman’s rank correlation on the two observed values for document length and the amount of redundant words. Between the character length of a tweet and redundancy we observed a correlation of  $\rho = 0.381$ . This indicates no or at most a very weak correlation. Also for the total number of words and the number of redundant words, we found a quite low value of  $\rho = 0.377$  allowing us to conclude that in a microblogging environment document length does not seem to be caused by verbosity.

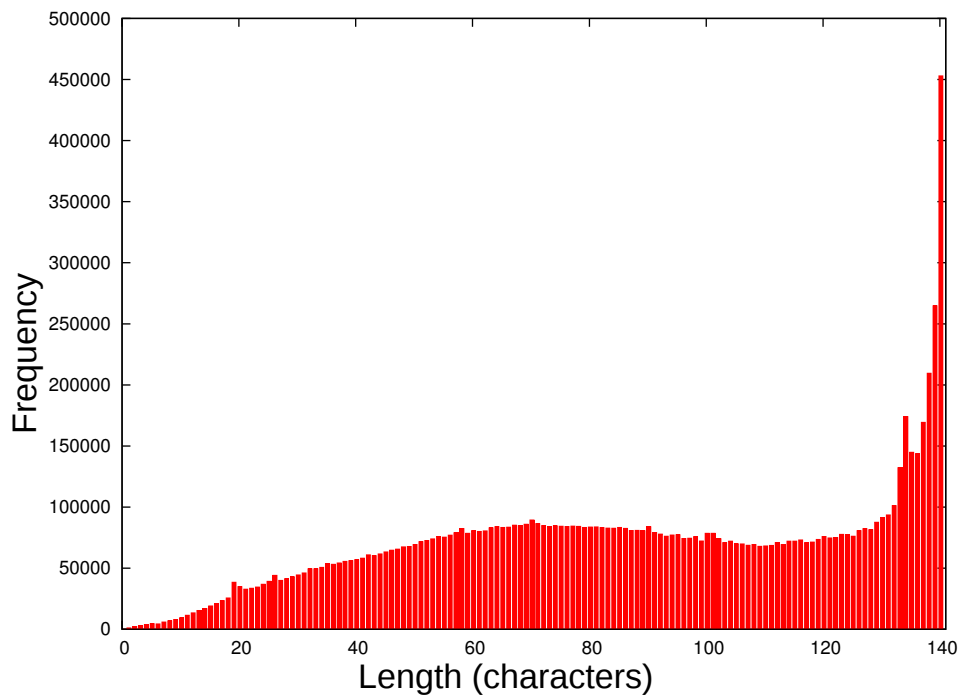


Figure 4.5: Distribution of document length (in characters) in the CHOUDHURY dataset.

A larger scope within a document is more difficult to detect. In order to get an idea of how many topics a tweet might cover, we applied latent Dirichlet allocation (LDA) to our dataset.

We used LDA to obtain 100 topics and the likelihood of each tweet to belong to this topic. Given this model, we found around 8.5% of the tweets

not being strongly related to any topic<sup>3</sup>. Among the remaining tweets, we analyzed the tweets for the number of topics that contribute to more than half of the probability of associated topics. To do so we analyzed how many topics we have to consider for each tweet to cover more than 50% of the probability in the topic mixture. We observed that in 77.1% of the tweets this point is already reached with one topic, meaning that the most prominent topic dominates the overall topic of the tweet. And for 99.6% the two top ranking topics contribute half the probability in the topic mixture. Our scope observations are also in line with the work of Zhao et al. [129] that observed that a single tweet usually covers a single topic. As a result, we can say that microblog messages in general are very focused – typically one LDA topic explains very well the composition of the entire tweet.

As neither the verbosity nor the scope hypothesis seem to apply to Twitter, we conjecture that length normalization for Twitter messages is not necessary. On the contrary, it might be counterproductive, as it introduces a bias favoring short over long messages without a justification.

#### 4.4.2 “Interestingness” as Static Quality Measure

Microblog documents differ from classical documents in quality aspects. Obviously, there is the distinction of spam, the trustworthiness or the purpose of the message. Quality is a static measure for a document that is independent from an actual query. Such a static quality measure is of particular interest when documents are likely to obtain homogeneous relevance values in a retrieval model. This is the case for Twitter, given that the term frequency is nearly a binary value on microblogs and queries are typically composed of very few or even single term.

In Section 4.1 we generalized the aspect of interestingness of a tweet as notion of quality, i.e. if the tweet would be of interest beyond the closer social neighborhood of the author. Thus, we consider a tweet to be of quality if it is interesting for readers. The motivation of introducing interestingness into a general retrieval setting is, that when searching for messages a user would want to leave his closer social sphere and get results also from other areas in the social graph. We base our notion of interestingness on the *retweet* function as described in Section 4.1. A user retweets when it finds a message particularly interesting and worth sharing with others. This notion of a tweet being potentially interesting to other users is then a suitable way to capture the static content quality we have in mind.

We follow the approach provided in the Section 4.3 to determine interestingness via the probability of a tweet being retweeted. We train a logistic regression model to be able to obtain for an individual tweet a probability of a retweet. We interpret this probability to be the quality of a microblog

---

<sup>3</sup>No topic had an association of more than 1%

message. If the probability of the retweet is high, the message is seen as interesting for a wider audience and, therefore, of better quality in a general retrieval scenario.

## 4.5 Applications and Evaluation

We apply and evaluate our approach of static content quality measure empirically in two different setups of microblog retrieval. In the first setup, we use the proposed measure for re-ranking (Section 4.6) of the tweets retrieved against the users' information need specified in the form of a query topic. To evaluate the approach in this setting, we used crowdsourcing mechanism provided by Amazon Mechanical Turk (MTurk). Results from this experiment show that introducing a static content quality measure in retrieval helps improve the ranking of microblogs in certain situations.

In the second setup, we participated in Microblog Track of TREC 2011, where the participating systems were required to include "interesting" tweets relevant a given query in the resultset. Participating in TREC 2011 helped us to test the performance of our approach on a large scale under variety of different settings and conditions in a real time microblog retrieval. The system setup and evaluation results are discussed in Section 4.7.

## 4.6 Ranking Microblogs

We empirically evaluate our approach in two different ways. We use a relevance-based evaluation following the classical Cranfield paradigm [23] and a subjective evaluation asking users which result sets they prefer for a given query.

### 4.6.1 Retrieval System Setup

To analyze empirically the impact of length normalization and message quality in the sense of interestingness on the retrieval performance, we set up three systems making use of two Lucene-based indices over a collection of 10 million tweets. Lucene<sup>4</sup> is a java based, open source, indexing and full-feature text search engine library suitable for most of the applications that require full-text search.

**lucene.** This system corresponds to an index based on out-of-the-box implementation of Lucene, i.e. a vector space retrieval model (VSM) including length normalization.

**lucene-noLen.** In this system Lucene index was modified not to perform length normalization, but otherwise use the Lucene retrieval function.

---

<sup>4</sup><http://lucene.apache.org/core/>

**retweet-odds.** In order to incorporate interestingness as static quality measure, we use an approach based on reranking relevant results. For this purpose we take the top-100 entries in a relevance-based result set and rerank them according to descending probability of retweet. This means rank 1 is the tweet that has the highest value for interestingness among the tweets with 100 highest relevance values.

For all setups we employ a simple duplicate detection mechanism for removing near-identical tweets from the search results using Jaccard’s coefficient. The Jaccard’s coefficient is used for measuring similarity between two sets or variables (binary and non-binary) and can be computed using Equation 4.2. Given the two sets of  $n$  members, Jaccard’s coefficient measures the overlap that both sets share with each other.

$$J(\mathcal{A}, \mathcal{B}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} \quad (4.2)$$

In our settings two messages were considered duplicate if their Jaccard’s coefficient on character 4–grams was equal or above the threshold level of 0.5. This might seem a rather low threshold at a first glance, however the choice of this threshold is attributed again to the short message length and the sparsity of data. We empirically verified on additional queries, that this setting performed well in removing duplicates and near duplicates.

#### 4.6.2 Evaluation Method

As corpus, we employed an existing dataset consisting of approximately 10 million tweets [21]. The tweets cover 118,506 users and were collected in the time between 2006 and 2009. The ratio of actual retweets in the dataset was 7.89%. The full list of datasets we used is given in Table 4.1.

To formulate information needs on Twitter, we run LDA on the Twitter dataset to obtain 100 topics. LDA returns for each topic a ranked list of terms based on their weight (probability) in the topic in a descending order of their weights. We randomly selected 20 topics from the 100 topics and took the top terms in each of the topic to formulate the queries. To cover different query lengths, we created five queries each of length one to four terms. In this way we could simulate a range of quite general and quite specific information needs, for which the data set should also provide relevant documents. Table 4.8 lists the queries we used for evaluation.

To assess the objective performance of retrieval on microblogs in a classical Cranfield setting, we needed objective relevance judgements. We applied pooling on the top 5 retrieved tweets for each method and evaluated the messages in the pool for relevance. As relevance is typically judged as *aboutness*, i.e. to which degree a document covers the topic of an information need, we

Table 4.8: Queries used to describe microblog information needs.

<b>1 word queries</b>	<b>2 word queries</b>
beer	iphone apple
coffee	gaza israel
weather	service health
photo	hair care
wii	wine price
<b>3 word queries</b>	<b>4 word queries</b>
video watch youtube	game watch play football
windows beta install	fashion beauty design dress
social media network	home kid mom wife
eat cook dinner	snow today ski cold
site web design	market search internet engine

additionally had the judges determine if a tweet was actually interesting. The purpose of this extension is to distinguish between technically relevant (i.e. about the topic, containing the query terms) and actually informative tweets (i.e. about the topic and providing general information on the topic). As we will see later quite often retrieval results on microblog messages are technically relevant, because they contain the query terms, but are practically not informative because they contain no other terms. Other tweets are rather personal messages, that do not satisfy a general information need.

To further measure user satisfaction, we set up a second experiment in which we confronted the assessors with two top-10 result lists for a given query, originating from two different retrieval setups anonymized as System A and System B. The users were asked if they preferred the results of System A, of System B, or were indifferent. The intention of this evaluation setup was to capture a subjective preference of a particular system over another one in direct comparison.

For both the evaluation tasks, we used Amazon Mechanical Turk (MTurk) to obtain relevance judgements and system preferences respectively. MTurk is a crowdsourcing platform that provides coordinated human intelligence for tasks which require large scale use of manpower. The requesters can post the tasks in MTurk and workers can browse the tasks and can participate if it satisfies the qualification criteria set by the requesters. Typically, a Requester can accept or reject the task completed by a Worker if it is not of a quality work. As, the motivation for the Workers to participate in the task is to earn money, therefore, there is a considerable chance that tasks are completed hastily to earn more money and raising the questions of work quality produced by Workers. There are few practices which are recommended in literature to obtain a quality work from MTurk. For ex-



ample, the Requester may setup a qualification test to verify the Worker qualification or the Requester can setup a trap by introducing the false or non-relevant results in the task, which the Requester can later use to filter out the spam results.

To assert the quality of our crowdsourcing approach we introduced artificial non-relevant results to identify and eliminate spammer results. Further, we collected for each evaluation task the feedback from seven different judgements and derived the final judgement based on the recommendations provided in [2]. Employing crowdsourcing for annotation [116], evaluation [2] and other manual tasks to support IR system evaluation has proven to be a suitable strategy for reducing time, cost and effort of such work.

### 4.6.3 Evaluation Results

We measure the performance of our approach using average P@5 and Mean Average Precision (MAP). The details of these measures have already been discussed in Chapter 2. Figure 4.6 gives the achieved values on all queries and a break down for the different query lengths. The plots list the Lucene system in its out-of-the-box configuration (i.e. a VSM with length normalization) as a baseline, the modified Lucene setup which does not perform any length normalization, and the setup using the retweet odds for reranking the top-100 results.

With respect to both evaluation metrics, we observe a global and local trend. Globally we can confirm our theory that length normalization on microblog messages is counterproductive. The standard Lucene approach is outperformed on all levels and independent of query length. Looking at the tweets retrieved, length normalization favors shorter messages. In particular for short queries, the tweets quite often consist only of the query terms, thus, not satisfying any information need. Turning off the length normalization bias leads to better results. The gap is narrower for longer queries, but on average, the approach without length normalization still leads to better results.

The second trend is depending on the query length. For short and under-specified queries a simple relevance-based ranking provides relatively poor results. Deactivating length normalization does improve the results, but not to a level that can be observed for longer queries. Here, incorporating interestingness as static quality measure leads to a big improvement. This is of particular interest, as queries on Twitter are typically short: 1.64 words on average [118].

Tables 4.9 and 4.10 demonstrate the positive effect of reranking based on interestingness with a very clear example of the query *beer*. The top ranking results in Table 4.9 contain only the query term, repeated many times. Technically these results are relevant as they contain the query term, but they do not convey much information. The results employing reranking

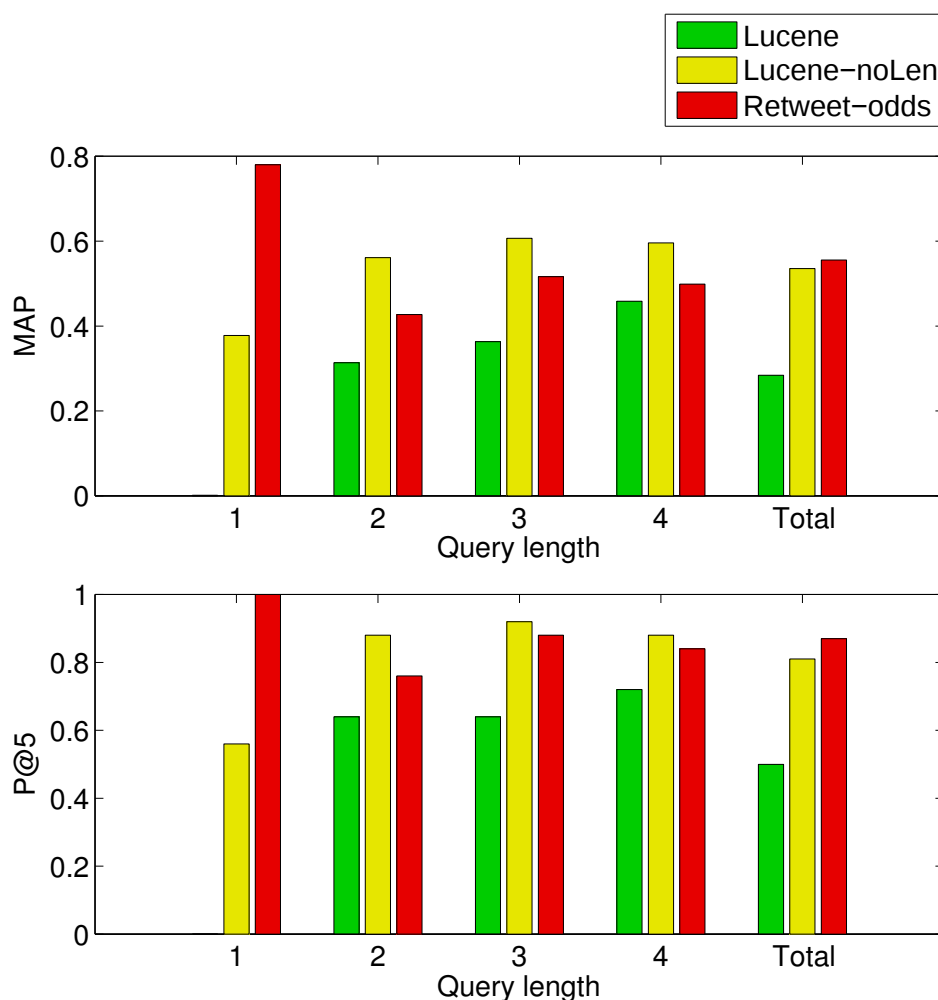


Figure 4.6: MAP and P@5 performance: in total and resolved by query length.

based on interestingness in Table 4.10, instead, are technically relevant and informative.

In general, with an average P@5 value of 1 and a MAP value of 0.78, retweet odds reranking on 1 term queries achieves the overall best results we observed in our evaluation. Looking again at the data we saw two explanations: (a) the top-100 results coming out of the relevance ranking were all more or less related to the topic, thus, reranking did not bring irrelevant documents to the top of the list and (b) the top relevance ranking results consisted mainly of tweets formed by the query term, as already noted above. The longer and more specific the queries are, the less the top relevance ranked documents consists of query terms alone and the more irrelevant tweets come into the top-100 documents used for reranking. Therefore, the

Table 4.9: Top 10 tweets for the query *beer* using the Lucene-noLen setting.

<i>Rank</i>	<i>Tweet</i>
1	Beer beer beer beer beer beer beer beer beer beer beer beer beer. Er, guess what I'm looking forward to?
2	BEER^5. RT @dewbelle: BEER BEER BEER BEER. RT @kulturbrille: BEER BEER BEER. RT @Bluebarrow: BEER BEER. RT @WalterMitty007: BEER
3	<a href="http://ping.fm/p/Bnra7">http://ping.fm/p/Bnra7</a> - In!!! BEER, BEER, BEER, BEER, BEER, BEER, BEER, BEER, BEER, BEER,
4	Lompoc. beer beer beer beer beer beer beer beer beer beer. <a href="http://twitpic.com/168ld">http://twitpic.com/168ld</a>
5	Beer, beer beer, beer, beer beer, beer and a little bit more beer.
6	beer beer beer beer beer beer beer. Simple 3pm
7	chickan and beer, chickan and beer, chickan and beer, chickan and beer, chickan and beer, chickan and beer: for brackfast?
8	Beer. Hot dog. Pickle. Beer. Hot dog. Pickle. Beer. Hot dog. Pickle. Beer. Hot dog. Pickle. Beer. Hot dog. Pickle. Beer. Hot dog. Pickle.
9	RT @grumpy_gardener Remember the true meaning of Memorial Day weekend. Beer, beer, beer, beer, beer!
10	New York City Beer Events - Beer Tasting - New York Beer Festivals - New York Craft Beer <a href="http://is.gd/39kXj">http://is.gd/39kXj</a> #beer

advantage of interestingness as static quality measure wears off and might actually put highly interesting, but only marginally relevant tweets to the top of the result lists.

Our second evaluation looked at the subjective performance. The assessors were presented with two result sets containing top 10 tweets for a given query from each system anonymized as System A and System B. The question asked from the assessors was which of the resultset the assessor prefers given the query. Also here we observed the same trends as above. Table 4.11 shows how many times a system was preferred over an opposed system. Figure 4.7 further summarizes these results and shows for each query length how many times the resultset of an individual system was preferred.

We can see the same tendency that the classical Lucene implementation is rarely preferred over another setup. Further we note strong preference towards the Retweet-odds based reranking on shorter and underspecified queries, while on longer queries the users prefer the relevance based ranking

Table 4.10: Top 10 tweets for query beer using the Retweet-Odds setting.

<i>Rank</i>	<i>Tweet</i>
1	UK beer mag declares "the end of beer writing." @StanHieronymus says not so in the US. <a href="http://bit.ly/424HRQ">http://bit.ly/424HRQ</a> #beer
2	beer summit @bspward @jhinderaker no one had billy beer? heehee #narm - beer summit @bspward @jhinde <a href="http://tinyurl.com/n29oxj">http://tinyurl.com/n29oxj</a>
3	Go green and turn those empty beer bottles into recycled beer glasses! — <a href="http://bit.ly/2src7F">http://bit.ly/2src7F</a> #beer #recycle (via: @td333)
4	Great Divide beer dinner @ Porter Beer Bar on 8/19 - \$45 for 3 courses + beer pairings. <a href="http://trunc.it/172wt">http://trunc.it/172wt</a>
5	Interesting Concept-Beer Petitions.com launches&hopes 2help craft beer drinkers enjoy beer they want @their fave pubs. <a href="http://bit.ly/11gJQN">http://bit.ly/11gJQN</a>
6	Beer Cheddar Soup: Dish number two in my famed beer dinner series is Beer Cheddar Soup. I hadn&#8217;t had too.. <a href="http://bit.ly/1diDdF">http://bit.ly/1diDdF</a>
7	New York City Beer Events - Beer Tasting - New York Beer Festivals - New York Craft Beer <a href="http://is.gd/39kXj">http://is.gd/39kXj</a> #beer
8	Love beer? Our member is trying to build up a new beer drinker's forum. Grab a #beer and join us: <a href="http://tr.im/pD1n">http://tr.im/pD1n</a>
9	#Baltimore Beer Week continues w/ a beer brkfst, beer pioneers luncheon, drink & donate event, beer tastings & more. <a href="http://ping.fm/VyTwg">http://ping.fm/VyTwg</a>
10	Seattle and Beer: I went to Seattle last weekend. It was my friend's stag - he likes beer - we drank beer.. <a href="http://tinyurl.com/cpb4n9">http://tinyurl.com/cpb4n9</a>

over the Retweet-odds reranking approach. This also corresponds to the observations made for the objective evaluation setting.

## 4.7 Filtering Microblogs

In our second evaluation, we participated in the Microblog Retrieval Track of TREC 2011. We adapted our LiveTweet system according to the guidelines<sup>5</sup>

<sup>5</sup><https://sites.google.com/site/microblogtrack/2011-guidelines>

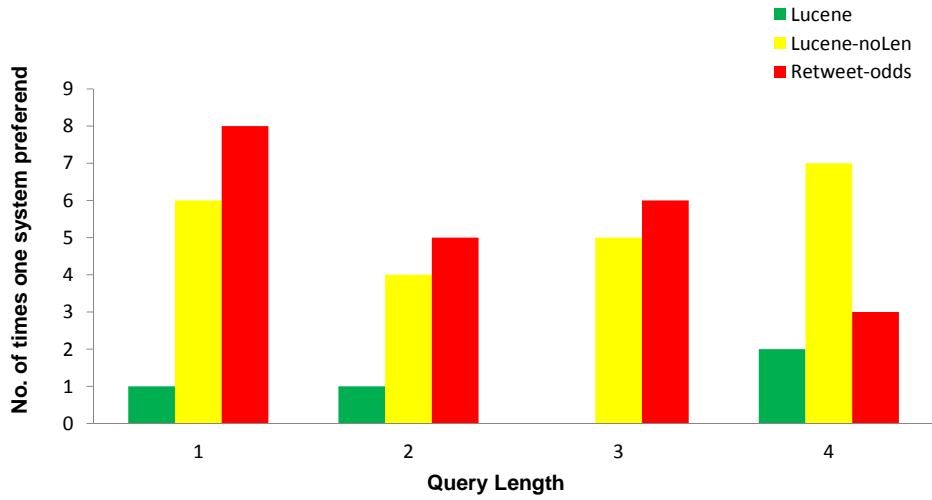


Figure 4.7: Results for number of times assessors' preferred the resultset from each participating system over the others for different query lengths.

Table 4.11: Results of number of times assessors preferred resultset from one system over the other systems.

	Opposed			Preferred system		
<b>Q1</b>	Lucene	Lucene-noLen	Retweet-odds	Lucene	Lucene-noLen	Retweet-odds
Lucene	–	4	5	–	4	5
Lucene-noLen	1	–	3	–	–	3
Retweet-odds	0	2	–	–	2	–
<b>Q2</b>	Lucene	Lucene-noLen	Retweet-odds	Lucene	Lucene-noLen	Retweet-odds
Lucene	–	3	2	–	3	2
Lucene-noLen	0	–	3	–	–	3
Retweet-odds	1	1	–	–	1	–
<b>Q3</b>	Lucene	Lucene-noLen	Retweet-odds	Lucene	Lucene-noLen	Retweet-odds
Lucene	–	2	4	–	2	4
Lucene-noLen	0	–	2	–	–	2
Retweet-odds	0	3	–	–	3	–
<b>Q4</b>	Lucene	Lucene-noLen	Retweet-odds	Lucene	Lucene-noLen	Retweet-odds
Lucene	–	3	2	–	3	2
Lucene-noLen	0	–	1	–	–	1
Retweet-odds	2	4	–	–	4	–

provided by Microblog Track which were formulated as real-time adhoc re-

trieval task. In this task the users' information need is represented by a query at a specific time. The user expects the system to provide most recent but relevant information to the query. Hence, the system should favor "interesting" but "newer" relevant tweets for a given query topic, ordered according to the chronology of the tweets with time starting at the time query was issued. Inclusion of the tweets' interestingness in relevance made the task as filtering: given the timestamp associated with the query, arrange all tweets in reverse chronological order and then throw away the non-relevant ones.

#### 4.7.1 Dataset

For this evaluation, we used the Twitter dataset provided by the TREC 2011 organizers. This Tweets11 corpus consists of tweets sampled from Twitter stream including both important and spam tweets. This corpus has a total of 16 million tweets including retweets as well. A total of 50 query topics were provided by the National Institute of Standards and Technology (NIST) to represent an information need for a given point in time. NIST provided the assessors to judge the relevance of the tweet for a given query. The lengthwise distribution of queries (number of words in each query) are provided in Table 4.12.

Table 4.12: Length-wise distribution of query topics against their frequency as provide by the TREC 2011 organizers to be used by the participating systems in the retrieval of tweets.

Query Length (in words)	1	2	3	4	5	6	7
Frequency	1	7	20	14	6	0	1

#### 4.7.2 Evaluation Criteria

The criteria used to judge a tweet as relevant in the TREC 2011 is that not only the tweet should be technically relevant to the query but it should also be "interesting". A tweet is considered relevant when it is topical and interesting (informative) for a given query topic. Interestingness is a subjective criteria, but the user might interpret it as providing additive value along with the relevance of tweet to the query. Interestingness is indeed subjective, but anyone who reads a tweet could still make a judgment on whether it is interesting or not, given the query topic.

The language used to express the information need was English and non-English tweets were deemed as non-relevant if present in the resultset. The assessors assessed the tweets on a three-point graded scale of relevance namely not-relevant, minimally-relevant, and highly-relevant. A tweet is

minimally relevant when it is about the topic but not sufficiently interesting. Highly relevant tweets are tweets which either contain highly informative content, or link to highly informative content. For example, for a topic about “Dublin”, a highly relevant tweet would tell something informative about Dublin or link to an informative news article, story etc. Minimally relevant tweets are non-retweet such as “dublin dubline dublin”. True retweets (starting with RT or so) were considered as non-informative and thus deemed as non-relevant, as the goal was to find highly informative tweets in an informational search settings and true retweets do not add any new information to the original tweet. If RT occurs somewhere inside the tweet but not at start (adding some information to original tweet), that was considered as relevant.

### 4.7.3 LiveTweet System

This was a real-time search task, the system was supposed to consider the collection as a stream of tweets and not as a static collection of tweets. Therefore, based on the features introduced in the Section 4.3.1 we train an incremental Naive Bayes model to obtain for an individual tweet the probability of retweet. In line with our findings in Section 4.3, we interpret this probability as the quality of a microblog message. If the probability of retweet is high, the message is seen as interesting for a wider audience and, therefore, of better quality in a general retrieval scenario.

The model is incremental with respect to the temporal order of the tweets in the dataset. This means that for a tweet at time  $t_i$ , we use the tweets up to time  $t_{i-1}$  to train our Naive Bayes classifier. We then apply this classifier to determine the likelihood of the tweet at time  $t_i$  to be retweeted and assign this value as a static quality measure to the tweet. Then we include this tweet’s features and the information whether it actually is a retweet into the classifiers knowledge base to update the prediction model for the next upcoming tweet at time  $t_{i+1}$ .

Given the limitation of the task to English tweets, we first use a language detection module to filter out all non-English tweets. The module is implemented using a dedicated language detection mechanism optimized for short texts [38]. We manually create a gold standard for English and non-English tweets on a small subset of 1,000 tweets from the given TREC corpus. After removing URLs, usernames and hashtags as well as reducing excessive repetitions of single characters (e.g., mapping *coooooool* to *cool*), we obtain a suitable accuracy of 96.9% at separating English from non-English tweets.

After filtering out the non-English tweets we compute the interestingness value of a tweet as defined in Section 4.3.1. Technically, our incremental Naive Bayes system assumes the presence and absence of features as results of a Bernoulli experiment with different a posteriori probabilities given we are observing an interesting (i.e., retweeted) or an uninteresting (i.e., not

retweeted) tweet. As incorporating sentiment detection requires external knowledge in the form of a dictionary annotated with sentiments, we operated the system once without sentiment features (run `WESTfilter`) and once with sentiment features (run `WESTfilext`).

In order to incorporate interestingness as static quality measure at retrieval time, we investigated two approaches: one based on filtering out non-interesting tweets, while maintaining a given ranking and one in which we additionally reranked the entries in a given result set according to their interestingness. For the purpose of filtering tweets of low interest we look at the relevant entries using a classical vector space model without length normalization. In this result set we look at the distribution of the interestingness values and identify a turning point in this distribution. We observed a general tendency of interestingness to decline fast after the most interesting tweet. Then, interestingness seems first to level out before starting again to drop more and more drastically. This turning point between the slowing and increasing decline in interestingness serves as a dynamic cutoff point (threshold  $t$ ) in our system. The remaining tweets are ranked according to their interestingness value.

TREC required to acknowledge and submit separate run of the system if it uses information that is external to the tweet contents for ranking. Computing the sentiment feature of a tweet requires the use of sentiment dictionary which is an external information to the tweet contents, therefore, we submitted separate versions of LiveTweet systems that incorporated tweet sentiment as a feature. Summarizing our approaches, we submitted the LiveTweet system in four different settings which are as follows:

**WESTfilter:** retrieving and ranking tweets by our modified VSM and then filtering out tweets having an interestingness less than the threshold  $t$ .

**WESTfilext:** retrieving and ranking tweets by our modified VSM and then filtering out tweets having interestingness less than the threshold  $t$ , but incorporated the sentiment of a tweet for computing its interestingness value.

**WESTrelint:** retrieval by the modified VSM, filtering out tweets having an interestingness less than the threshold  $t$  and finally re-ranking the tweets by their interestingness score.

**WESTrlex:** retrieval by the modified VSM, filtering out tweets having an interestingness less than the threshold  $t$  and finally re-ranking the tweets by their interestingness score. Again, here we incorporated the sentiment of a tweet for computing its interestingness value.



#### 4.7.4 Evaluation Results

The official metric used by TREC 2011 for evaluating the effectiveness of systems in the retrieval scenario was P@30 in a tweet-ordered ranking. However, participating groups were encouraged to analyze their systems using other measures as well. In particular, TREC 2011 provided four scenarios for evaluation:

**allrel** The official evaluation scenario corresponds to a filtering task on a stream of incoming messages. Thus, the ranking of messages is provided by the time at which the tweets in the result set were produced. New tweets are ranked higher, older tweets are ranked lower. The actual challenge for the retrieval system is to filter out all irrelevant tweets from the incoming stream.

**highrel** For a subset of the topics, the relevance judgments distinguished between relevant and highly relevant tweets. While otherwise equivalent to **allrel** the **highrel** evaluation scenario considered only the highly relevant tweets as actually relevant.

**by-score** Different from the two previous scenarios, here the task is evaluated as a classical retrieval scenario. This means, that for each topic, the system can actually provide a ranking of the relevant tweets. As in classic TREC evaluation for such a setting, the ranking is imposed by the ordering the documents according to the relevance score provided by the system.

**by-rank** Additionally the TREC Microblog guidelines allowed to provide a ranking which diverged from the actual order imposed by the score. We used this freedom to use the ranking of a VSM for the tweets combined with a filter retaining only highly interesting messages. This means the ranking is imposed by a classical retrieval model, but some tweets were discarded from the result set.

We used MAP, nDCG, P@5, P@10, P@20, P@30, R-Prec and bpref for **allrel**, **highrel**, **by-score** and **by-rank** scenarios to compute the performance of all four variants of the LiveTweet system. As stated above, the relevance of a tweet was judged on a graded scale. This graded relevance judgments distinguish between the **allrel** and **highrel** evaluation scenarios and we also used it to compute nDCG.

Figure 4.8 shows the performance of LiveTweet for different measures using the **allrel** evaluation scenario. In **allrel**, we do not see significant difference in the performance as the runs based on filtering and re-ranking provide the same resultset and the ranking is implied by the timestamps of the tweets. So, in **allrel** it is only of interest to compare between using or not

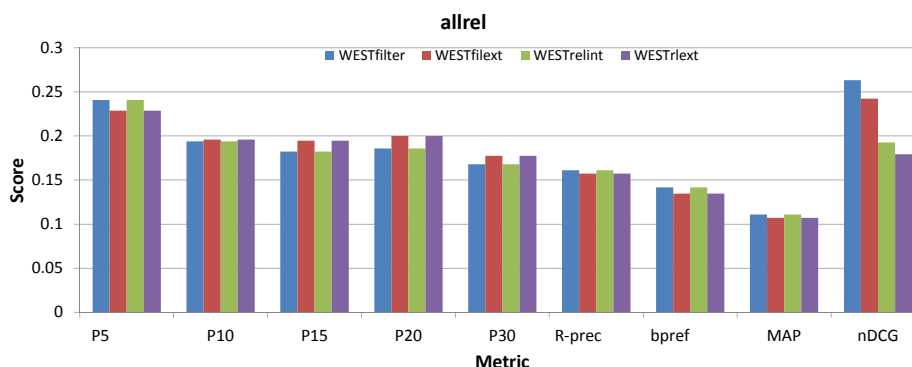


Figure 4.8: Evaluation results of LiveTweet at **allrel** scenario under various performance measures.

using external knowledge. While there is a small decline in the performance when introducing external knowledge, it is not of statistical significance.

Figure 4.9 shows the performance of LiveTweet for various measures using **by-rank** evaluation scenario, which corresponds more to a retrieval scenario. We see again that the performance of the runs using or not using external knowledge does not differ significantly from each other across all the evaluation measures. Thus, the runs that are actually of interest for comparison are WESTfilter and WESTrelint. The best performance is achieved by the WESTfilter across all measures. Here, the observed improvements in performance are statistically significant.

Table 4.13 summarizes the results and provides information about significance of the improvements for **allrel** and **by-rank** between WESTfilter–WESTfilext and WESTfilter–WESTrelint. From the results we conclude that interestingness is more suitable to be incorporated as a filter function; re-ranking the results according to interestingness demonstrated a poorer performance.

Figure 4.10 finally shows the performance of LiveTweet variants over individual query topics under the by-rank evaluation scenario. Looking at individual topics gives additional insights, when considering the length of the actual query.

It has been observed that Web queries have an average length of 3.08 words, while on Twitter the average query has only 1.64 words [118]. In Table 4.12 we provide an overview of the frequency distribution of the query topics in the TREC Microblog track with respect to the length of the query measured in words. This distribution is in favor of longer queries which are more representative for general Web search but seem to be less typical for Twitter search.

Figure 4.11, Figure 4.12, Figure 4.13 and Figure 4.14 show the MAP performance of LiveTweet System with respect to number of terms in a query

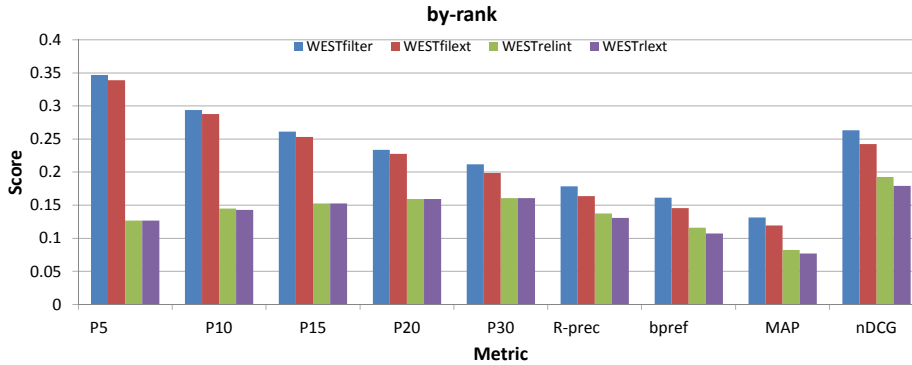


Figure 4.9: Mean average precision (MAP) scores of LiveTweet at **by-rank** scenario.

Table 4.13: Statistical test of significance between the performance of **WEST-filter** and **WESTfilext** at **allrel** and **by-rank** scenarios for various measures.

	allrel			by-rank		
	WESTfilter	WESTfilext	Sig*	WESTfilter	WESTrelint	Sig*
P@5	0.2408	0.2285	—	0.3469	0.1265	***
P@10	0.1939	0.1959	—	0.2939	0.1449	***
P@15	0.1823	0.1946	—	0.2612	0.1524	***
P@20	0.1857	0.2	—	0.2337	0.1591	**
P@30	0.168	0.1775	—	0.2116	0.1605	**
MAP	0.1109	0.1071	—	0.1312	0.0822	***
bpref	0.1416	0.1347	—	0.1612	0.1159	***

\*]— not significant, \* significant at 5%, \*\* significant at 1%, \*\*\* significant at 0.1%

using allrel, highrel, by-score and by-rank evaluation scenarios respectively. In all of the four variants of the system we see a negative correlation between the query length and MAP performance of the system. As indicated in Section 4.4, using interestingness is particularly useful for short queries, as they are typical for Twitter [118]. The exception to this observation is highrel scenario, where there is no document retrieved for one word queries. This is due to the fact that in gold standard dataset provided by the TREC 33 queries out of 49 have documents annotated as highly relevant by the assessors. For one word queries, there are no relevant documents in the gold standard and that lead to a zero MAP score in our evaluation under highrel scenario. We also checked the correlation between the mean average precision and the length of the queries measured by the number of terms. We observe a strong negative correlation of  $-0.967$  which hints in the direction

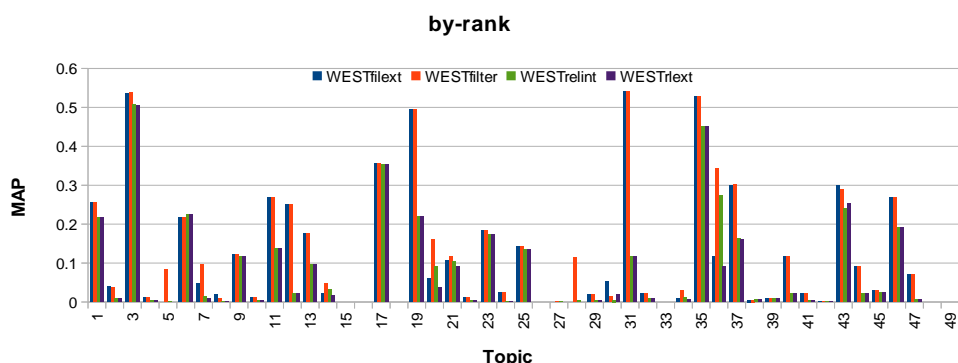


Figure 4.10: Mean average precision (MAP) score of LiveTweet for individual topics at **by-rank** scenario.

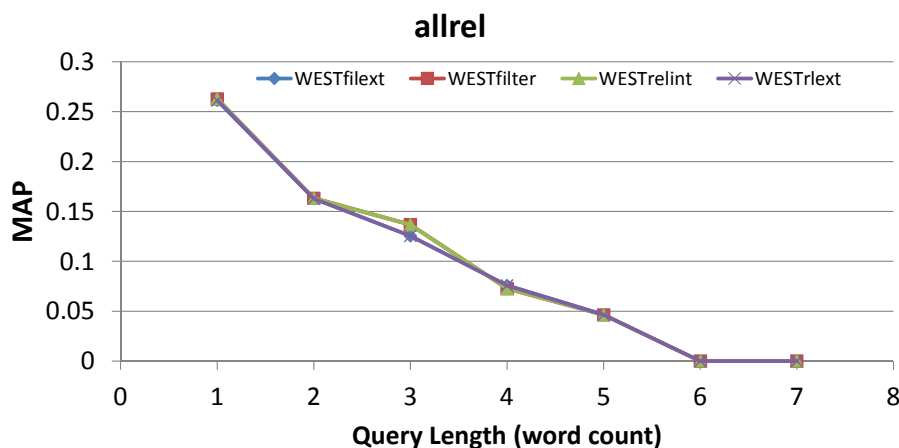


Figure 4.11: Mean average precision (MAP) score of LiveTweet under individual query groups at **allrel** scenario.

that, as observed in previous work, the model used in LiveTweet actually performs better on short queries.

## 4.8 Related Work

Twitter has become the focus of much research in recent years. Thus, in this section we concentrate on work covering the design or adaptation of retrieval models for searching in microblogs, twitter content analysis, and static quality metrics such as influence, interestingness or user status.

People seek information within microblogs in two ways: by asking questions to their followers or by querying over microblogs in order to discover information that has already been posted [31, 118]. This finding is also

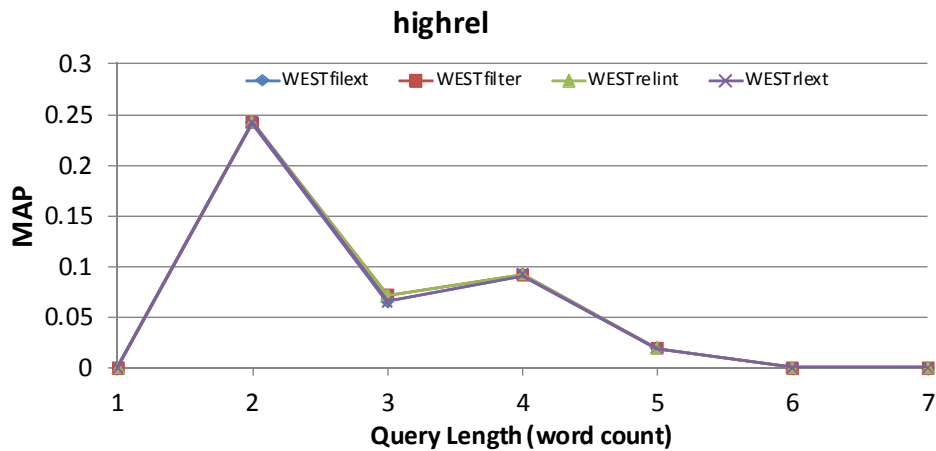


Figure 4.12: Mean average precision (MAP) of LiveTweet against query length for by-rank.

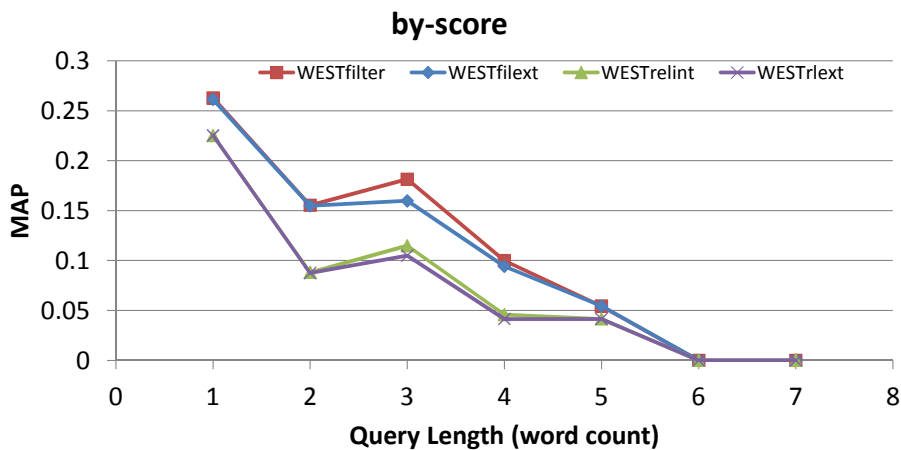


Figure 4.13: Mean average precision (MAP) of LiveTweet against query length for by-rank.

supported by Java et al. [56] that identified “information sources” and “information seekers” as main categories for users on Twitter. Work by Che Alhadi et al. [18] further differentiates the intended purposes of “information source” users when posting a single tweet.

Looking at the users’ side, people mainly search microblogs to find in particular timely information (e.g. news, trending topics, events), social information (information related to other users) and topical information (e.g. topic of interest) [118]. Another observed and important difference between search on microblogs and on the web is the length of queries. While web

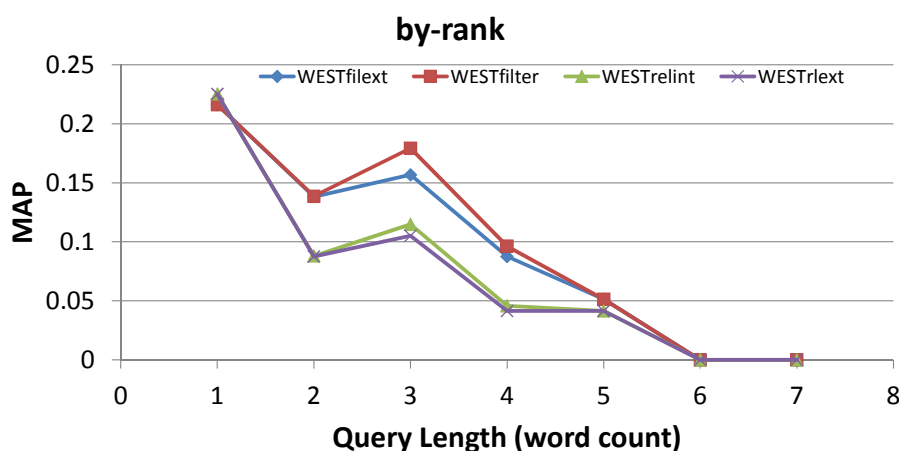


Figure 4.14: Mean average precision (MAP) of LiveTweet against query length for by-rank.

queries are on average 3.08 words long, queries on microblogs are far shorter: on average 1.64 word. Sakaki et al. [107] investigated the real-time nature of Twitter, in particular for event detection. They used Twitter to build an earthquake reporting system in Japan and notify registered users about such events by sending them emails. The system outperforms the Japan Meteorological Agency in speed of notification.

Other works used metadata (hashtags) for enhancing topical retrieval in microblogs. A recent study by Huang et al. [51] analyzes the tagging practice in Twitter and finds that users add tags to their messages in Twitter to join discussions on existing topics. Tagging in Twitter is intended for filtering and directing content in order to make it appear in certain streams.

Previous studies have also measured the influence of users on tweet quality. Cha et al. [13] and Romero et al. [104] discovered that a very large number of followers do not necessarily have an impact on a user being retweeted more often. This indicates that the popularity of a user does not automatically imply a higher influence or authority in Twitter. It also shows that the current ranking in Twitter based on the number of followers or on in-degree alone is not enough to find the most influential users.

However, some researchers [49, 69, 81, 97, 117] found that the context of a tweet (number of followers/followees, timestamp) has strong correlation with retweetability. These analyses also consider a small set of content features: the presence of URLs and hashtags is a strong influential factor for a tweet to be retweeted.

Hong et al. [49] measure the popularity of messages based on the number of retweet and use machine learning technique to predict how often new messages will be retweeted. The content of messages, temporal infor-

mation, metadata of messages and users, and the user's social graph were used as features in predicting whether messages will get retweeted. Kwak et al. [69] used three different measures to identify influential users on Twitter. They ranked the users by the number of followers, PageRank and number of retweets. As a result, they found that the ranking of the users based on the number of followers and PageRank are very similar, while rankings based on the number of retweeted messages is different, concluding that interest does not necessarily correlate with social status.

Cha et al. [13] also used three different measure; number of followers, number of mentions and number of retweets. They disagree that social network features such as a large numbers of followers is correlated with the likelihood of a user's messages to get retweeted. Hence, the social status is not sufficient as a static quality measure to indicate authors that will provide interesting information to their followers.

A study on predicting a tweet to be retweeted was shown in work by [49, 97]. Hong et al. [49] use retweets as a measure of popularity and apply machine learning techniques to predict how often new messages will be retweeted. The authors analyze the content of messages, temporal information, metadata of messages and users, and the user's social graph as the features in predicting the messages to be retweeted. Petrović et al. [97] carry out human experiments for the task of deciding if a tweet will be retweeted. They applied a passive-aggressive algorithm to automatically predict tweets as human. They also used social and content features of the tweets as quality indicator for predicting the retweet.

The current form of Twitter's own search function does not perform optimal, when it comes to rank interesting tweets at the top of the result list. As a remedy, Massoudi et al. [59] presented an approach incorporating query expansion and quality indicators into a retrieval model for searching microblog posts for a given topic of interest. They used emoticons, post length, shouting, capitalization, hyperlinks, reposts, followers, and recency as the quality indicators. Twitter trending topics with an average length of 1.4 words were used to collect almost 110 millions tweets. With respect to this approach our method is different as we analyze a wider, different and purely content based set of features to derive a quality indicator.

In analogy with PageRank, Weng et al. [126] define the TwitterRank measure to rate users. Nagmoti et al. [81] state that social graph network features (number of followers and followees) can be used as a ranking measure of microblog search. Although these methods may be used to predict the popularity of a tweet, they cannot be used as a rank for finding high-quality tweets, as they are based on user rankings and contextual information instead of content.

Some researchers [49, 69, 81] mentioned that, in addition to content features, the social status strongly correlates with the likelihood of a tweet to be retweeted and, thus, to have a wider reach. Nagmoti et al. [81] considered

social network properties of the authors (e.g. the number of followers and followees or the number of posted tweets) to rank microblogs posts. They also used the tweet length and the presence of a URL as a quality measure of interestingness and informativeness of the information shared with other.

In summary, These most recent works indicate that the likelihood of a tweet to be retweeted is based on the context of the tweet (number of followers or followees, time of the tweet, age of the account) and elementary features of the content of a tweet ( presence of URL's, hashtags, trending topics). All these approaches use social context of the user (social graph). We explicitly drop the context information from the analysis as our emphasis is not on who get retweeted or who writes interesting tweets. Instead we put much stronger emphasis on the content and analyze a wider set of low-level content-based features as well as derived, high-level content-based features (topics and sentiment of the tweet). So, there is no straight forward comparison between our approach and approaches who use social context and other non-content information for prediction task. Our work also focuses on the applicability of a content-based probability of retweet as a static quality measure.

## 4.9 Summary

In this chapter we looked into particularities of information retrieval on microblogs: document quality and sparsity. We analyzed microblog contents and introduced a method to determine the quality of a microblog message using the notion of interestingness, and evaluated the method in two different ways in an information retrieval task on microblogs. Further, we showed that sparsity is inherent to microblog messages, as it reflects the technical constraints on the length of message. The quality of a document with respect to its ability to satisfy an information need originates from the different purpose and environment in which microblog messages are generated. We motivate from theoretical and data analytic point of view, that document length normalization introduces an unmotivated bias towards short documents in microblog retrieval.

To determine the interestingness of a microblog message, we based our method on the retweet function of Twitter as a measure for messages with a wider interest. To overcome the context bias of, e.g. user's social network, we used a learning approach on pure content features to predict the probability of a message to be retweeted. To capture the content we used low-level features such as presence of URLs, hashtags, usernames, question and exclamation marks, emoticons, positive and negative words, as well as high-level features such as sentiments and latent topics.

We made the following observations about the retweeting behavior of Twitter users: As a general rule, a tweet is likely to be retweeted when it



is about a general, public topic instead of a narrow, personal topic. For instance, a tweet is unlikely to be retweeted when it is addressed to another Twitter user directly, while our topic analysis revealed that general topics affecting many users like social media or Christmas are more likely to be retweeted. This can be understood as the Twitter platform being better suited as a news and announcement channel rather than a personal communication platform, complementing the description of Twitter as news media in [69]. A further interesting observation in this context is the tendency that bad news seem to travel fast in Twitter.

Finally, we introduced a way to use interestingness as static quality measure for microblog messages. We evaluated our approach in two different information retrieval tasks with the objective to measure the usefulness of the approach in re-ranking and filtering microblogs retrieved by modified vector space model. We empirically showed that this approach improves retrieval performance in the sense of providing more relevant and generally interesting messages in the search results. We noticed that our method achieved best performance when user's information needs are specified in the form of short and underspecified queries, which are typical for searching information in Twitter.



## Chapter 5

# Feature Sentiment Diversification of User Generated Contents: The FREuD Approach

Web 2.0 provides an interactive way for online text publishing in various domains. Users can engage in online discussion on a wide range of topics and contribute their personal experiences and opinions. One such area is online product review portals such as `reviews.cnet.com` and `epinion.com`. These portals do not only publish editorial reviews of different products but also provide ways for the users to share their own experience of the use of the products. In these reviews users tend to cover different aspects or features of the products. Usually a review covers some features of the product along with associated sentiments or opinions about these features. Usually a review covers some features of the product along with associated sentiments or opinions about these features. Other than being positive or negative about features, users also discuss which features are more important than others and about which features they are more excited. Thus, these reviews provide rich information about different aspects of a product and can play an important role in the decision making process of customers when buying a product.

Consider, for instance, a scenario where a customer intends to buy a smartphone. Nowadays, his first step would likely be to use a product review website and browse through user reviews for different smartphones to get an overview of the users' opinions about the usefulness of different features of each of the phones. The user reviews typically are neither structured nor constrained by a specific format or template. Therefore, users are free to express their experiences and opinions in free form text. A common observation is that successful products can have hundreds of reviews, where

some reviews are more useful than others. Another observation is that the reviews are written in free form text and the extent of coverage is different towards different features with different degrees of authority or authenticity. To arrive at some decision of whether or not to buy the phone the customer has to contemplate a large amount of text to find the most valuable reviews or opinions; which requires a lot of reading and time. Thus, the challenge in this scenario is to come up with an optimal set of high quality reviews that cover as many relevant features as possible and provide diversified view points of opinions of different users about the product features.

Currently, there are some ways to indicate which reviews are worth reading. One common way is to leverage user votes for reviews which indicate the usefulness of the review as seen by other users. But user votes do not guarantee that highly rated reviews cover all possible aspects and associated sentiments and that all the pros and cons are addressed. In fact, the collective dynamics may even lead to disproportionately high ratings of some, rather arbitrary, reviews (cf. the analogy with preferred downloads in [108]).

Automatic solutions could overcome these problems, but mining sentiments about product aspects or features from user reviews poses certain challenges. The first challenge is to estimate which features are addressed in a review. The second difficulty is to mine the users' sentiments. And the third challenge is how to come up with an optimal set of reviews, which has already been shown in literature to be a NP-hard problem [120].

To tackle the above mentioned problems, we consider this problem as an information retrieval task with specific emphasis on result diversification. Based on this mind-set, we developed the FREuD approach. In FREuD, we use a combination of text pre-processing and probabilistic topic models to obtain latent topics discussed in a collection of reviews related to a single product. Given the application scenario of FREuD in this work, we pre-processed the dataset in a way which when used in topic models is more useful for discovering specific topics than using the topic models on plain text. We observed, that these latent topics frequently align very well with the features of the product discussed in the reviews. Thus, the latent topics provide us with a very good approximation of the product features. We then employ a dictionary based approach for estimating the user sentiments in each single review. Finally, we select a subset of reviews to optimize the diversity criteria of covered product features and sentiments. To this end, we use a greedy algorithm operating on the features and sentiments discussed in each review.

The work in this chapter make two main contributions:

- We describe the problem setting of feature-centric sentiment diversification as an information retrieval task and present the details of our FREuD approach and discuss its technical details. We demonstrate that FREuD outperforms two baselines of a user based ranking as

it is currently implemented in productive systems as well as a naive sentiment diversification strategy.

- We develop a novel gold standard dataset for the task of feature-centric sentiment diversification over product reviews. To this end, we have had human assessors annotate a gold standard on the features covered and sentiments expressed about these features in reviews for twenty products in three different categories.

The rest of this chapter is organized as follows. In Section 5.1, we proceed with a formalization of the task of feature-centric sentiment diversification and its interpretation from an information retrieval viewpoint. Our approach is discussed in Section 5.2. The evaluation methodology, the employed data set and the construction of a gold standard is presented in Section 5.3, while the results and comparison of FREuD with baseline methods are given in Section 5.4. Section 5.5 covers previous work in the related areas. At the end, we conclude with a summary in Section 5.6 of the chapter.

## 5.1 Formal Task Definition

Before going in the details of our approach, we formalize the feature-centric sentiment diversification task we are addressing. The aim of this  $\text{FSCOVERAGE}(k)$  task is to generate a selection of  $k$  product reviews that cover as many product relevant features as possible and diversified range of sentiments over the features.

Let us consider a product  $\mathcal{P}$ . The set of reviews related and relevant to this product  $\mathcal{P}$  forms a corpus  $\mathcal{C}$ . This corpus  $\mathcal{C}$  constitutes the set of documents  $\text{FSCOVERAGE}(k)$  operates on. Furthermore, we consider a product  $\mathcal{P}$  (e.g. a mobile phone) to be associated with a finite set

$$\mathcal{F} := \{f_1, f_2, f_3, \dots, f_n\}$$

of product relevant features (e.g. screen size, battery life time, usability, etc.). Finally, we define a set of sentiment dimensions

$$\mathcal{S} := \{s_1, s_2, s_3, \dots, s_m\}$$

such as, positive, negative, calm, excited, etc. In the  $\text{FSCOVERAGE}(k)$  task, product features  $\mathcal{F}$  and sentiment dimensions  $\mathcal{S}$  define the space we want to cover as extensive as possible with a fixed number of reviews.

We can assume that the reviews in corpus  $\mathcal{C}$  address the features and utter sentiments about them to various degrees. It is possible that a review expresses both, positive and negative, sentiments about a certain feature, e.g. in a statement like “Multitasking is nice, but not all apps work yet”.

In review  $d$  we capture the strength of a positive sentiment  $s \in S$  regarding feature  $f \in \mathcal{F}$  with the value

$$v^+(f, s, d) \in [0, 1],$$

and the negative sentiment with

$$v^-(f, s, d) \in [0, 1]$$

Higher values of  $v^+(f, s, d)$  and  $v^-(f, s, d)$  correspond to stronger positive and negative sentiments, respectively. A value of 0, instead, means that no positive or negative sentiment is uttered about feature  $f$ .

For a given set  $\mathcal{C}' \subset \mathcal{C}$  of reviews, we can now define a feature-sentiment-diversity score  $Div(\mathcal{C}')$  as given in Equation 5.1.

$$Div(\mathcal{C}') = \sum_{f \in \mathcal{F}} \sum_{s \in S} \left( \max_{d \in \mathcal{C}'} v^+(f, s, d) + \max_{d \in \mathcal{C}'} v^-(f, s, d) \right) \quad (5.1)$$

The FSCOVERAGE( $k$ ) task is to maximize the score  $Div(\mathcal{C}')$  under the constraint  $|\mathcal{C}'| \leq k$ . In analogy to the proof in [120] it can be shown that FSCOVERAGE( $k$ ) is NP-hard.

## 5.2 The FREuD approach

We have already stated above the three main challenges for obtaining a feature-centric sentiment diversified selection of reviews. Using the formalization in the previous section we can state these challenges more precisely:

- Identify the set of features  $\mathcal{F}$  discussed in a set of reviews  $\mathcal{C}$ .
- Estimate the positive and negative sentiment values  $v^+(f, s, d)$  and  $v^-(f, s, d)$  for a given feature  $f$  in a specific document  $d \in \mathcal{C}$ .
- Provide a good approximative solution for FSCOVERAGE( $k$ ) based on these estimates.

To counter these challenges, we have developed the FREuD approach which combines machine learning techniques for product feature mining, a dictionary based approach for estimating the sentiments of a review and a greedy approach to provide a solution for FSCOVERAGE( $k$ ). We will now present the details for each of these steps in the following subsections.

### 5.2.1 Feature Extraction

To obtain a list of features discussed in a set of reviews, we use Latent Dirichlet Allocation (c.f. Chapter 3). Given the strong focus of the reviews and the overall scenario, we found that the LDA topics align very well with product features, thus, providing us with the set  $A$ .

To refine the input for LDA, we use the Stanford Part-Of-Speech Tagger [119] to extract nouns from the reviews. Most of these nouns reflect the different aspects of products. In LDA, for each product category we set the number of topics to be 10. This number approximates the number of features we use for each category in the evaluation dataset. Table 5.1 shows top terms of topics discovered by LDA which approximate to features discussed in reviews about cellphones, cameras and printers .

Thus, the latent topics  $\mathcal{Z}$  provide us with a very good approximation of the set of discussed product features  $\mathcal{F}$ . Accordingly we can consider each topic  $z \in \mathcal{Z}$  to correspond to a feature  $f \in \mathcal{F}$ . Furthermore, the topic composition of each review gives us an estimate to which degree a review discusses a specific feature. In conclusion, we use the probability  $p(f|d)$  as value for modeling the extent to which review  $d$  addresses feature  $f$ .

### 5.2.2 Sentiment Estimation

To estimate the sentiments in a review we employ the Affective Norms for English Words (ANEW) sentiment dictionary [8]. The emotional rating values for words in ANEW cover a range between 1 and 10. We normalize these values to the interval  $[-1, 1]$  and distinguish between the positive and negative values for the purpose of obtaining  $v^+(s, w)$  and  $v^-(s, w)$  for each individual word  $w$  and sentiment  $s$ . The global positive sentiment value  $v^+(s, d)$  of an entire review  $d$  is then given by an aggregation of the positive sentiment values of the single words as given in Equation 5.2.

$$v^+(s, d) := \sum_{w \in d} v^+(s, w) \quad (5.2)$$

The value for  $v^-(s, d)$  is defined equivalently.

### 5.2.3 Feature-Sentiment Estimation

To estimate the positive and negative sentiment  $s$  for a given feature  $f$  under each sentiment category in a review we use Equation 5.3 to combine the positive and negative global sentiment of a review  $d$  with the probability of the review to address feature  $f$  according to outcome of the LDA analysis.

$$v^+(f, s, d) := v^+(s, d) \cdot p(f|d) \quad (5.3)$$

$$v^-(f, s, d) := v^-(s, d) \cdot p(f|d) \quad (5.4)$$

Table 5.1: Topics as determined by LDA from the product reviews approximating the features of Camera, Cellphone and Printer category.

Category	Topic No.	Top Topic terms
Camera	1	lens, mode, iso, fps, value
	2	camera, image, quality, picture, dslr
	3	camera, video, quality, lens, slot
	4	control, sensor, pixel, zoom, issue
	5	photo, control, flash, option, set
	6	zoom, panason, grip, focus, inch
	7	camera, feature, body, kit, frame
	8	nikon, focus, issue, lcd, meter
	9	camera, water, shock, fog, claim
	10	perform, review, viewfinder, comparison
Cellphone	1	phone, problem, antenna, apple, case
	2	battery, speed, day, internet, wifi
	3	wifi, signal, custom, reception, gps
	4	phone, screen, quality, web, text
	5	phone, camera, device, quality, feature
	6	phone, lte, verizon, sens, data
	7	phone, video, screen, camera, atrix
	8	blue-tooth, sync, contact, voice, keyboard
	9	camera, feature, motion-blur , review, email
	10	bionic, size, device, razr, user, experience
Printer	1	toner, printer , cartridge, color, cost
	2	printer, color, laser, print, time
	3	printer, print, duplex, quality, laser-jet
	4	model, size, network, easy, setup
	5	quality, print, unit, time, streak
	6	fax, print, quality, window, color
	7	scan, epson, fax, busy, photo
	8	ink, cartridge, time, refill, price
	9	printer, paper, jam, card, tray
	10	photo, copy, canon, machine, line

#### 5.2.4 Review Subset Selection

As already mentioned  $FSCOVERAGE(k)$  is NP-hard. Therefore, to find a good solution for  $FSCOVERAGE(k)$  we use a greedy algorithm. The greedy algorithm starts from an empty result set  $\mathcal{R}$  of selected reviews and iteratively adds a review that adds most value to the result set in the sense, that it extends the range of covered features and expressed sentiments most.



The degree to which a combination of sentiment  $s$  and feature  $f$  is already covered in the result set  $\mathcal{R}$  is given by

$$\max_{d' \in \mathcal{R}} v^+(f, s, d') \text{ and } \max_{d' \in \mathcal{R}} v^-(f, s, d')$$

for the positive and negative sentiment values respectively. The gain of adding document  $d$  to the result set corresponds to the subsequent increase of these two maxima. By summing up the increase over all combinations of sentiments and features we obtain a contribution score  $contrib(d)$  for document  $d$ :

$$\begin{aligned} contrib(d) := & \\ & \sum_{s \in \mathcal{S}} \sum_{f \in F} \left[ \max \left( 0, \left( v^+(f, s, d) - \max_{d' \in \mathcal{R}} v^+(f, s, d') \right) \right) \right. \\ & \left. + \max \left( 0, \left( v^-(f, s, d) - \max_{d' \in \mathcal{R}} v^-(f, s, d') \right) \right) \right] \end{aligned}$$

After having computed this score for all documents which have not been added to the result set so far, we select the one review  $d$  with the highest  $contrib(d)$  score for addition to the result set  $\mathcal{R}$ . In the next iteration we recalculate the  $contrib(d)$  scores of the remaining reviews to determine which review to add next. This iteration is repeated until the set  $\mathcal{R}$  contains  $k$  reviews.

### 5.2.5 FREuD Variations

Looking at real world data, we observed few interesting properties about the dataset which are listed below.

- Our first observation is that there is a near linear relationship between review length and sentiment scores of the review. However, we also observed that there is a lot of variance in the strength of the sentiment expressed in the reviews. Figure 5.1 shows this relationship for valence, arousal and dominance.
- We calculated the number of sentiment words in each review and found that users tend to use more positive words than negative words in longer reviews as shown in Figure 5.2.
- We computed the ratio of positive to negative sentiment scores in a review and related it with the review length. We observed that this ratio is always above zero for reviews having total word counts greater than 50 as shown in Figure 5.3, implying that longer reviews usually discuss more positive aspects and will always be classified as positive on average.

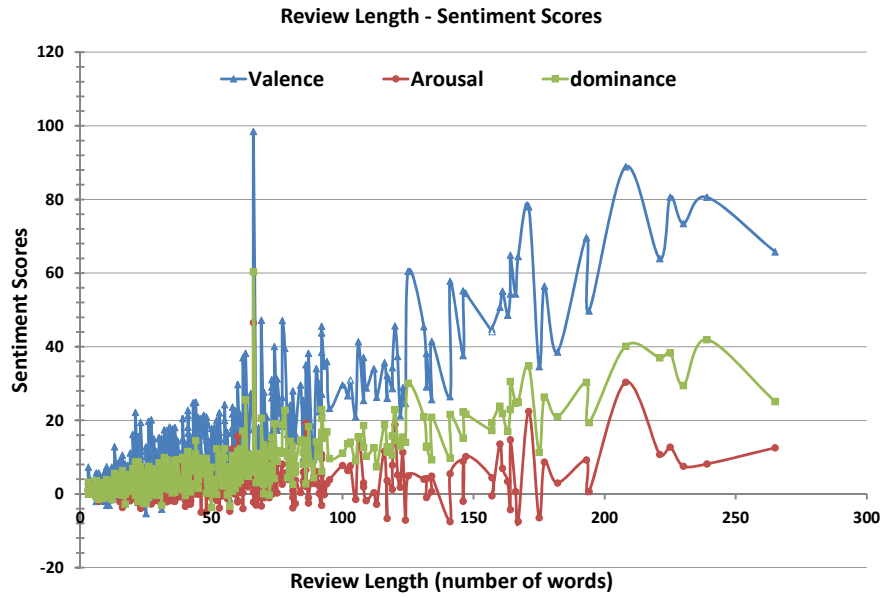


Figure 5.1: Plot showing relationship between review length measured in number of words and associated sentiment score.

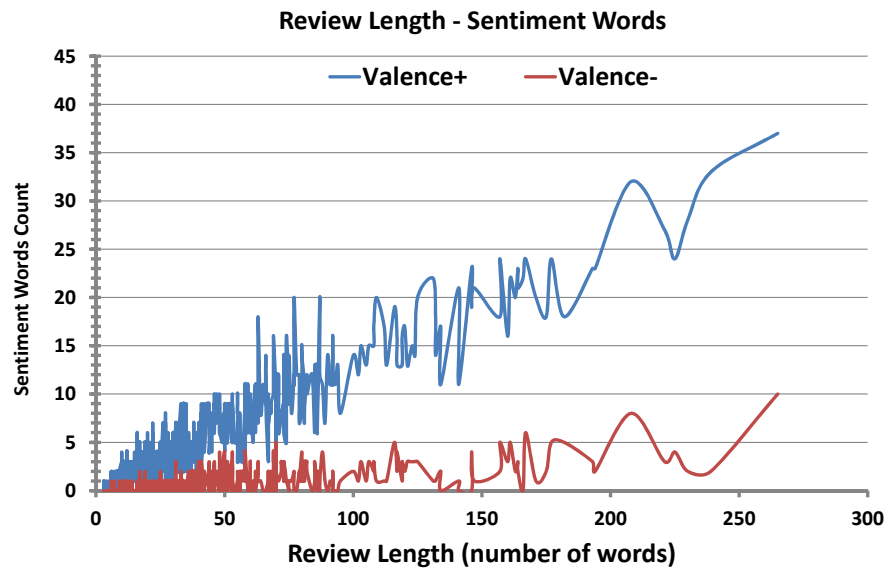


Figure 5.2: Plot showing relationship between review length and sentiment words used in the review .

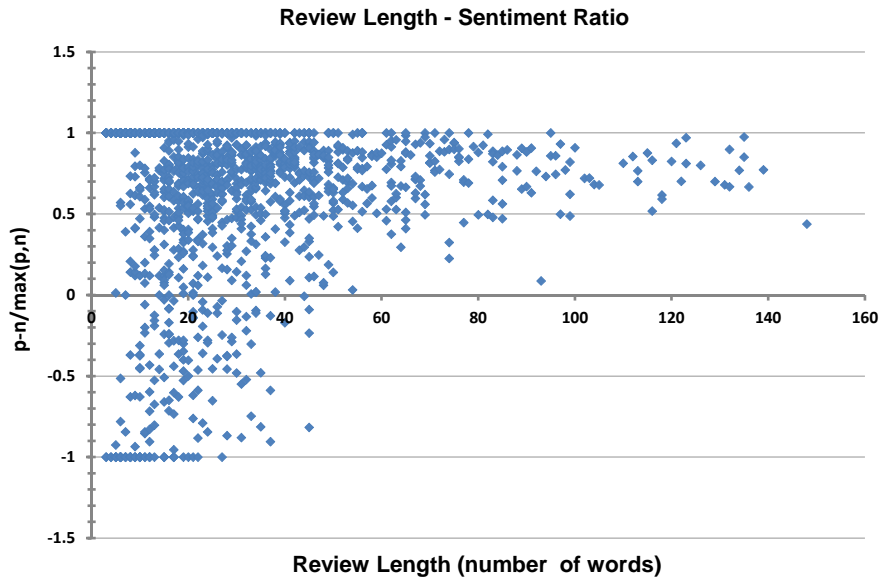


Figure 5.3: Plot showing relationship between review length measured in number of words to the ratio of positive (p) to negative (n) sentiment scores of the review.

Because of this imbalance in favor of positive scores, a longer review will always be classified as positive in our dataset. and this variance cannot be explained by the document length alone. The relationship between review length and sentiment scores has an effect on the  $contrib(d)$  function, which would favour longer documents. Therefore, to check the effect this bias has on the performance of our approach, we implemented FREuD in three different variations dealing with length normalization in different ways:

- **FREuD-noLN** : This variation does not make use of any length normalization technique for sentiment scores.
- **FREuD-stdLN** : In this implementation we use a standard length normalization for sentiment scores. We divide the global sentiment score of a review by the total number of words in the review.
- **FREuD-sentiLN** : Length normalization is performed based on the number of sentiment words in the review, i.e. the number of words which have actually been annotated with a sentiment score according to the ANEW dictionary.

Table 5.2: CNET product review dataset used for evaluating the FREuD approach.

Category	# Products	# Reviews
Cell Phone	7	1501
Printer	7	688
Camera	6	256

### 5.3 Evaluation Setup

In this section, we elaborate on the evaluation methodology for our FREuD approach. Our evaluation approach includes an objective evaluation using established metrics for measuring the performance of search result diversification systems. We describe the compilation of a data set and gold standard suitable for evaluating approaches on feature-centric sentiment diversified selection of reviews. In this context we also introduce the two baseline systems which we use for comparison.

#### 5.3.1 Dataset

As evaluation corpus we use end user product reviews collected from the CNET product review website<sup>1</sup>. CNET covers several product categories of consumer electronics. This website allows end users to write reviews about products and provide an overall rating of the products using a five-star rating system. The users also have the option to vote for the usefulness of existing reviews using a thumbs up and thumbs down voting system. By default, CNET uses these votes to rank the reviews in the user interface from the most helpful to the least helpful review.

We use the API of CNET<sup>2</sup> for obtaining and downloading product information and reviews about popular products under three categories: printers, cell phones and digital cameras. In each of the product category, we chose up to seven products with at least 40 reviews. We then crawled all the user reviews from these products along with their metadata, e.g. the star rating and the number of thumbs up and thumbs down votes for each review. In total we obtained 2,445 product reviews on 20 products for our evaluation data set. Except for the category of cameras each product had more than 50 reviews. Table 5.2 gives an overview of the corpus, while Figure 5.4 shows the distribution of the review corpus with respect to length in words.

<sup>1</sup><http://reviews.cnet.com/>

<sup>2</sup><http://developer.cnet.com/>

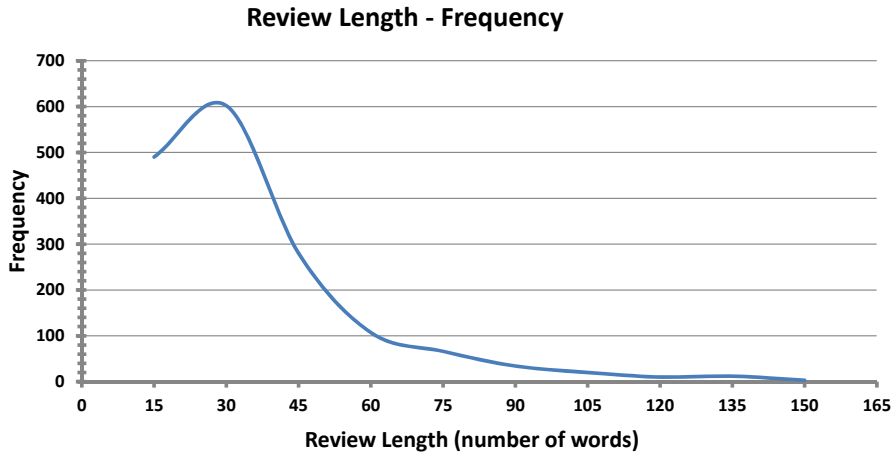


Figure 5.4: Plot showing length-wise distribution of reviews in our experimental dataset.

### 5.3.2 Baseline Systems

In order to judge the quality of our FREuD approach we need realistic baseline systems to compare to. As mentioned above, CNET by default ranks the reviews for each product on the basis of the helpfulness of the review. The helpfulness is computed on the basis of the thumbs up and thumbs down votes for the reviews. Such an approach is also implemented in many other product review portals. We reconstruct this ranking from the metadata of the reviews and use this approach as our first baseline system: **CNET-default**. Furthermore, as CNET displays five reviews per page and as in a Web context only few user go beyond the first page [54], we used a size of  $k = 5$  for the set of reviews for all approaches. This baseline allows for comparing to a realistic scenario implemented in productive, real world systems.

As we are interested in selecting a set of sentiment diversified reviews covering the positive and negative sentiments of the reviewers about the product, therefore, we require a baseline consisting of set of reviews that have both positive and negative sentiments about the product in them to compare against our FREuD approach. Our second **CNET-diversified** baseline implements a naive sentiment diversification strategy. As mentioned above, each product review in CNET provides also a star rating on a scale between 1 and 5 stars. If a user assigns 5 stars to the product, this implies he is highly positive about the product while 1 star means he is highly unsatisfied with it. For our **CNET-diversified** baseline, we pick one review from each of the five star rating categories. As there are typically multiple reviews with the same star rating, we always chose the one with the highest

usefulness score according to the thumbs up votes.

### 5.3.3 Developing a Gold Standard for the Dataset

We are interested in diversity based on both: sentiments and features. As the dataset does not directly exhibit objective and machine readable information about the covered features or expressed sentiments, we needed to obtain a gold standard in a different way. To this end, we first collected a list of product features for each of the categories and then employed crowdsourcing in order to obtain human feedback on whether or not a feature is discussed in a review and what are the sentiments about this feature.

The preliminary requirement for our evaluation setup was to obtain a list of typical product features for each product category. For cell phones, we obtained a list of features from *gsmarena*<sup>3</sup>, which uses a predefined structured list of features for cell phone comparisons. For digital cameras we used an equivalent list of features employed on *dpreview*<sup>4</sup> for reviewing cameras. Finally, we used three printer websites to collect the most commonly discussed features for the printer category. Table 5.3 lists the category wise features collected as mentioned above.

Table 5.3: List of features for Camera, Cellphone and Printer category collected from various websites to be used in the evaluation setup.

Category	No.	Feature
Camera	1	Battery (battery life, battery type, battery replace ability, time it take to recharge)
	2	Photo Quality (Megapixels, color accuracy, light balance)
	3	Video Quality (VGA, HD)
	4	Focus (auto focus, manual focus, accuracy, speed)
	5	Ease of Use (ergonomics, control buttons, user interface)
	6	Body (body size, body weight, body material)
	7	Zoom (digital zoom, optical zoom)
	8	Shooting speed (single shot, burst mode)
	9	Flash (flash range, external flash)
	10	Exposure (ISO levels, picture noise)
	11	View Finder (live view, optical view finder, information available in view finder)
	1	Camera (front and/or back camera, video quality, photo quality)

<sup>3</sup><http://www.gsmarena.com/>

<sup>4</sup><http://www.dpreview.com>

Cellphone

	2	Display Readability (contrast, brightness, reflection)
	3	Screen Resolution (resolution size, number of pixels per inch)
	4	Performance (responsiveness, speed, processing power)
	5	Design (look and feel, body material)
	6	Portability (weight, size)
	7	Battery (battery life, battery type, battery replace ability, time it take to recharge)
	8	Network Connectivity (2G, 3G, 4G/LTE, Wi-Fi, Bluetooth, tethering)
	9	Storage (internal and/or external)
	10	Availability of Applications
	11	Ease of Use (how easy to operate, accessibility of functions)
	12	Ease of Use (how easy to operate, accessibility of functions)
	13	Music (playback, sound quality)
	<hr/>	
Printer	1	Printing Speed
	2	Text Print Quality
	3	Photo Print Quality
	4	Running Cost (price of toner/cartridge, number of pages printed per toner/cartridge)
	5	Duplex Printing (auto duplex, manual duplex)
	6	Document Feeder (paper tray size, paper size and type, paper capacity)
	7	Network Connectivity (LAN connection, Wi-Fi Connection)
	8	Ease of Use (operating interface, initial printer setup, replacing inks)
	9	Operating Noise
	10	Scanner (resolution, scanning speed)
	11	Fax
	12	Copier
<hr/>		

In the next step we had all five approaches (CNET-default, CNET-diversified, FREuD-noLN, FREuD-stdLN and FREuD-sentiLN) compute a set of top-5 reviews for each product. We pooled the reviews obtained in this way and had them evaluated by human assessors in a crowdsourcing fashion. The assessors were then asked to mark which of the features from the given lists were covered in a review and which sentiments were expressed about these features. The dataset used for this evaluation is shown in Table 5.4.

The reviews for each product in each category were mixed and anonymized for systems names. The assessors were not aware of the originating system of the review.

Table 5.4: Dataset annotated by the assessors for features and sentiments to be use in the evaluation.

Category	# Products	# Reviews	# Features
Cell Phone	7	175	13
Camera	6	150	11
Printer	7	175	12

The human assessors were iteratively asked to pick a product category and select a product in that category for which they wanted to read a review and provide details on which features are discussed in the review and which sentiments are expressed about these features. The sentiment choices available to assessors for selection were ‘positive’, ‘negative’, ‘neutral’ and ‘both’. The option ‘both’ meant that a reviewer is positive and negative for a given feature. Assessors could evaluate any number of reviews. An identification of the assessors avoided that the same assessor could work on the same review twice. Assessors’ prior use and knowledge of the products was also recorded. To collect the assessors feedback on the review, we used the process as described by Algorithm 2.

There were 179 unique assessors who voluntarily participated in the evaluation<sup>5</sup>. Each of the review in the evaluation was presented to three different assessors. A feature is deemed as covered in the review if 2 out of 3 assessors agreed that the given feature was discussed. For the sentiment polarity of the feature we employed a similar majority decision. Table 5.5 shows some details of the participating evaluators.

Further details of the gold standard dataset and website used for obtaining annotations are provided in Appendix A.

### 5.3.4 Inter-rater Agreement

To gain confidence in our gold standard, we checked the inter-rater agreement among the evaluators over the feature and sentiment coverage they identified. For this purpose we used Fleiss Kappa [34]. Fleiss Kappa is a widely used inter-rater reliability measure employed to check for nominal scale agreement between a fixed number of raters. The product category wise inter-rater agreement over the coverage of a feature in a review is shown in Table 5.6, which according to the Fleiss benchmark [33] is an *intermediate*

<sup>5</sup>A large share of the evaluation was completed by research fellows from various research group who were typically interested in developing such a gold standard dataset to be used later in other experiments.



```

assessor selects a product;
for each unassessed remaining review do
  randomly pick one review at a time and present it to the assessor
  side by side with the product specific preselected features;
  assessor reads review and ;
  for each feature (from the list): assessor checks do
    if feature is discussed in the review at all then
      for all found utterances discussing this feature do
        assessor ticks the appropriate option (positive, neutral,
        negative, both) to annotate the sentiment and polarity
        of the feature;
        mark the location of the utterance in the review;
      end
    else
      go to next feature;
    end
  end
end

```

**Algorithm 2:** Process used to obtain assessors feedback while developing gold standard.

Table 5.5: Statistics of the assessors who participated in the annotation process to obtain gold standard dataset.

<b>Gender</b>	<b>Percentage</b>
Males	68.72%
Females	29.05%
Undisclosed	2.23%

<b>Product Knowledge</b>	<b>Percentage</b>
No	51.79%
Little	29.91%
Yes	18.30%

Table 5.6: Category-wise inter-rater agreement among assessors over coverage of the features in the reviews.

Category	Fleiss Kappa ( $k$ )
Printer	0.45
Cell Phone	0.44
Digital Camera	0.41

Table 5.7: Category-wise inter-rater agreement among assessors over feature-sentiment annotations in the reviews.

Category	Fleiss Kappa ( $k$ )
Printers	0.34
Cell Phones	0.33
Digital Cameras	0.31

to good agreement for all the categories. While, inter-rater agreement about the sentiment annotation of a feature is given in Table 5.7, which is a *fair* agreement according to Fleiss benchmark.

### 5.3.5 Diversity Evaluation Metrics

Measuring the performance of algorithms which combine relevance and diversity together requires metrics which can incorporate relevance and diversity in a ranked retrieval evaluation setup. One established metric is  $\alpha$ -nDCG (c.f. Chapter 2) which builds on standard nDCG. The assumption underlying  $\alpha$ -nDCG is that each query has multiple known intents or facets and these intents are of equal importance. The  $\alpha$ -nDCG metric regards the documents in a result set to cover these query intents to different degrees. A highly relevant document is one which covers many intents. Additionally,  $\alpha$ -nDCG promotes an increase in diversity by reducing redundancy.

In our settings, we can assume each product to serve as query and product features as different known intents of the query with each intent (feature) having equal likelihood or importance. For a given product, we consider each feature-sentiment pair as one intent and if a review covers more such intents, it should be ranked higher than the others. In our settings, we used the standard value of  $\alpha$  used also in the TREC diversity task, i.e.  $\alpha = 0.5$ .

For computing the diversification performance of the different approaches, we used the TREC evaluation framework provided for the diversity task of the Web Track<sup>6</sup>. We generated appropriate input files (*qrels*, *topics* and *results*) for the TREC tool from our gold standard dataset and the result

<sup>6</sup><http://plg.uwaterloo.ca/~trecweb/2009.html>

files from the rankings provided by all the competing approaches. In our gold standard dataset, we also counted distinct feature–sentiment pairs that were covered in the top-5 reviews for each product category. We found that gold standard dataset covers all possible feature–sentiment pairs under each product category. Given our setting and the motivation described above we cut off the result list for all approaches after five results. Accordingly we compare the performance based on  $\alpha$ -nDCG@5.

## 5.4 Experimental Results

In this section, we present and discuss the experimental results from the evaluation of the FREuD variations and the baseline approaches in selecting the sentiment diversified top-5 reviews. As mentioned in the Section 5.3.5, we measure the performance based on the  $\alpha$ -nDCG@5 metric. Tables 5.8 to 5.10 compare the  $\alpha$ -nDCG@5 scores for all approaches and for each individual product in the Camera, Cellphone and Printer category respectively. We have highlighted the best performing approach for each product. We see that FREuD-noLN and FREuD-sentiLN achieve high scores in general and dominate the baseline approaches for most products. FREuD-stdLN still provides very good results in some cases, but the values are less stable and exhibit a larger variation.

Table 5.8: Diversification performance comparison of all approaches under individual products in **Camera** category using  $\alpha$ -nDCG@5.

System	Products						
	1	2	3	4	5	6	7
<b>Freud-noLN</b>	0.80	<b>0.84</b>	<b>0.76</b>	0.74	<b>0.80</b>	0.86	0.80
<b>Freud-sentiLN</b>	<b>0.82</b>	0.81	0.66	<b>0.87</b>	0.73	0.85	<b>0.85</b>
<b>Freud-stdLN</b>	0.39	0.63	0.66	0.87	0.79	0.77	0.77
<b>CNET-default</b>	0.72	0.68	0.63	0.86	0.67	<b>0.93</b>	0.73
<b>CNET-diversified</b>	0.31	0.72	0.47	0.77	0.77	0.71	0.83

This behaviour is also reflected when considering the average performance of the approaches. Figure 5.5 shows the overall average  $\alpha$ -nDCG values. Here we observe that FREuD-noLN dominates all other systems including the two baseline systems as well as the other two variations of FREuD. However, the values of 0.74 and 0.72 for FREuD-noLN and FREuD-sentiLN, respectively, are very close to each other. For FREuD-stdLN, instead, we see that the average performance is actually below the CNET-default baseline. The naive sentiment-diversification of CNET-diversified performs worst. The poor performance of FREuD-stdLN can be explained by the fact that standard length normalization favors the short length re-

Table 5.9: Diversification performance comparison of all approaches under individual products in **Cellphone** category using  $\alpha$ -nDCG@5.

System	Products						
	1	2	3	4	5	6	7
<b>Freud-noLN</b>	<b>0.75</b>	0.57	<b>0.77</b>	0.64	0.64	<b>0.82</b>	0.84
<b>Freud-sentiLN</b>	0.63	0.47	0.48	<b>0.71</b>	<b>0.90</b>	0.60	<b>0.91</b>
<b>Freud-stdLN</b>	0.70	<b>0.74</b>	0.57	0.29	0.78	0.92	0.42
<b>CNET-default</b>	0.72	0.50	0.74	0.65	0.88	0.44	0.84
<b>CNET-diversified</b>	0.37	0.67	0.31	0.31	0.81	0.61	0.40

Table 5.10: Diversification performance comparison of all approaches under individual products in **Printer** category using  $\alpha$ -nDCG@5.

System	Products					
	1	2	3	4	5	6
<b>Freud-noLN</b>	<b>0.65</b>	0.72	<b>0.77</b>	<b>0.77</b>	0.50	0.79
<b>Freud-sentiLN</b>	0.57	<b>0.76</b>	0.73	0.43	<b>0.83</b>	<b>0.87</b>
<b>Freud-stdLN</b>	0.63	0.59	0.22	0.43	0.33	0.80
<b>CNET-default</b>	0.47	0.60	0.61	0.47	0.50	0.79
<b>CNET-diversified</b>	0.49	0.45	0.55	0.70	0.30	0.85

views to be ranked higher as their length normalized sentiment scores are higher than the sentiment scores of longer reviews. As shorter reviews typically cover a lower number of features, therefore, the collective feature coverage of the reviews recommend by FREuD-stdLN is less than the other two FREuD approaches resulting in low  $\alpha$ -nDCG values for FREuD-stdLN.

Table 5.11 illustrates the relative improvement in  $\alpha$ -nDCG@5 scores achieved by FREuD variations over the two baselines. Also in this case we see a noticeable gain in performance by FREuD-noLN and FREuD-sentiLN.

Table 5.11: Relative percentage improvement in  $\alpha$ -nDCG scores achieved by FREuD variations against the two baseline systems.

<b>FREuD Variations</b>	<b>CNET-default</b>	<b>CNET-diversified</b>
Freud-noLN	10.52%	29.99%
Freud-sentiLN	8.02%	27.05%
Freud-stdLN	-8.24%	7.92%

To check whether the difference in performance is significant, we conducted a paired t-test on the  $\alpha$ -nDCG@5 scores. The results are show in Table 5.12. We see that FREuD-noLN and FREuD-sentiLN performed sig-

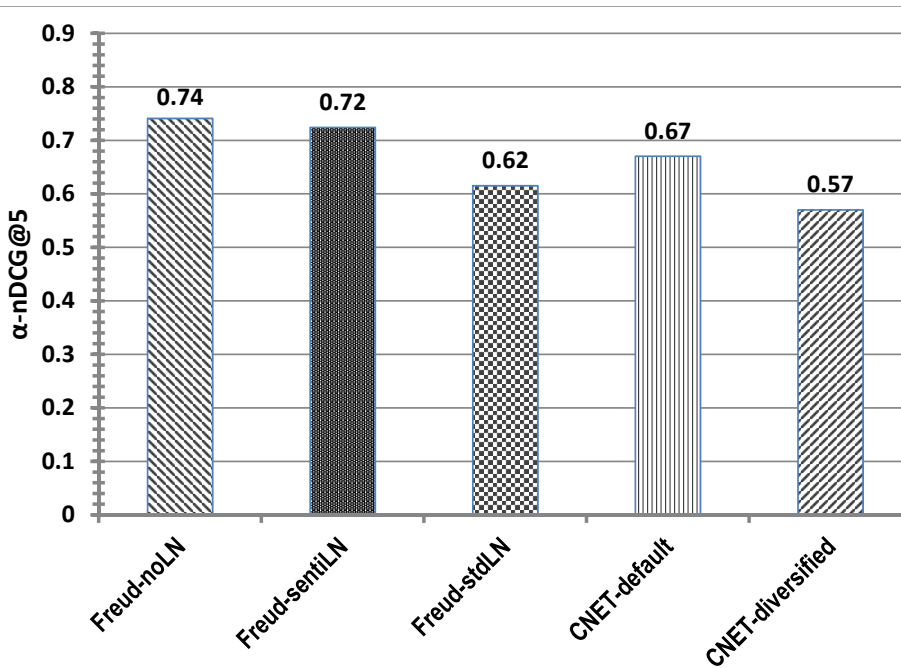


Figure 5.5: Plot showing review diversification performance of all approaches aggregated for each product category.

nificantly better over the two baselines at 5% significance level. While the performance difference of FREuD-stdLN against the baselines is not significant.

Table 5.12: Results showing statistical significance of differences in performance of FREuD and baseline systems using t-test at 5% significance level.

<b>FREuD Variations</b>	<b>CNET-default</b>	<b>CNET-diversified</b>
Freud-noLN	*	*
Freud-sentiLN	*	*
Freud-stdLN	-	-

To analyse differences in the different product categories, we computed the performance of all approaches at product category level. Figure 5.6 and Table 5.13 shows the category wise average  $\alpha$ -nDCG@5 scores. Here we observe the same trend as before, i.e. also category-wise FREuD-noLN dominates all other systems when it comes to coverage and diversity performance. In the category-wise split-up we also see that the performance of FREuD-noLN and FREuD-sentiLN is at par for the categories camera and printer, while for cell phones FREuD-noLN has minor advantage.

Table 5.14 shows the relative percentage improvement of FREuD varia-

Table 5.13: Diversification performance comparison of all approaches using average  $\alpha$ -nDCG@5. under each individual product category.

System	Camera	Cellphone	Printer
<b>Freud-noLN</b>	<b>0.80</b>	<b>0.72</b>	<b>0.70</b>
<b>Freud-sentiLN</b>	<b>0.80</b>	0.67	<b>0.70</b>
<b>Freud-stdLN</b>	0.70	0.63	0.50
<b>CNET-default</b>	0.75	0.68	0.57
<b>CNET-diversified</b>	0.65	0.50	0.56

tions over CNET-default baseline. We see that FREuD-noLN improves over the CNET-default in all three categories, while FREuD-sentiLN improves in Camera and Printer categories. The performance of FREuD-stdLN is always below the CNET-default, the reason for which has already been explained above. Table 5.15 shows percentage improvement of FREuD variations over the CNET-diversified baseline. Here we see that other than one instance in Printer category all FREuD variations achieved improvement over the CNET-diversified baseline.

Table 5.14: Relative percentage improvement in diversification achieved by FREuD variations over the **CNET-default** baseline.

System	Camera	Cellphone	Printers
Freud-noLN	7.24%	5.70%	22.20%
Freud-sentiLN	7.04%	-1.31%	22.49%
Freud-stdLN	-6.43%	-7.12%	-12.57%

Furthermore, we also tested the category-wise differences in performance for significance. These results are reported in Table 5.16 against CNET-default and in Table 5.17 against CNET-diversified at 5% significance level. Compared to CNET-default, a significant difference is observed only in the printer category for FREuD-noLN and FREuD-sentiLN. All other differences are not significant<sup>7</sup>. Compared to CNET-diversified, FREuD-noLN

Table 5.15: Relative percentage improvement in diversification achieved by FREuD variations over the **CNET-diversified** baseline.

System	Camera	Cellphone	Printers
Freud-noLN	22.27%	44.43%	25.48%
Freud-sentiLN	22.04%	34.84%	25.77%
Freud-stdLN	6.69%	26.90%	-10.23%

<sup>7</sup>Compared to the global performance, this can be explained with the smaller sample

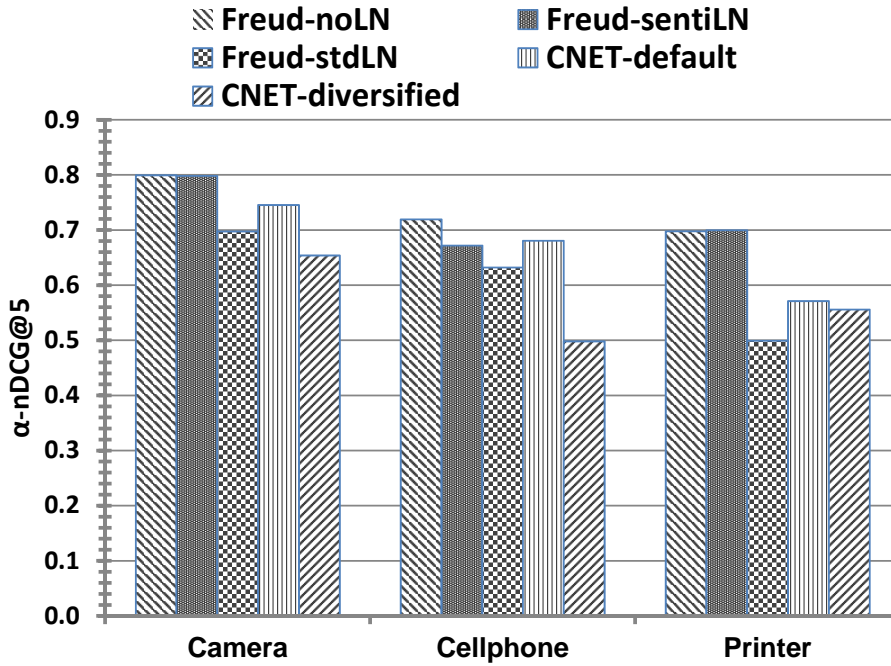


Figure 5.6: Plot showing review diversification performance of all approaches for each individual product category.

showed a significant improvement in all three categories and FREuD-sentiLN performed significantly better in the categories camera and printer.

Table 5.16: Category-wise statistical significance test of performance difference against *CNET-default* (at 5% significance level).

System	Camera	Cellphone	Printers
Freud-noLN	-	-	*
Freud-sentiLN	-	-	*
Freud-stdLN	-	-	-

Concluding our experiments, we can clearly see that FREuD-noLN performs best and significantly outperforms the baseline algorithms in selecting feature-centric sentiment diversified reviews. The improvement is consistent over the several product categories and significant at a global level. We can further say that length normalization of sentiment scores using sentiment word count in a review does not have significant effect on the performance of FREuD. However, standard length normalization of sentiment scores deteriorate the performance when compared with other two FREuD variations size which makes it harder to demonstrate statistical significance.

Table 5.17: Category-wise statistical significance test of performance difference against *CNET-diversified* (at 5% significance level).

System	Camera	Cellphone	Printers
Freud-noLN	*	*	*
Freud-sentiLN	*	-	*
Freud-stdLN	-	*	-

as well as CNET-default baseline.

## 5.5 Related Work

Our work in this chapter mainly relates to two areas of research: text mining and diversity ranking. Therefore, we specifically focus on mining product features, estimating sentiments from free text in an unsupervised way and using this information for product review diversification. So, in this section we concentrate on related work in these areas.

Feature extraction techniques mainly rely on the availability of structured or semi-structured documents. Guo et al. [44], for instance, proposed an unsupervised product-feature extraction and categorization method from semi-structured reviews. Their method relies on extracting features mentioned explicitly in structurally indicated pros and cons sections in a review. Liu et al. [74] proposed a supervised method for detecting product features in semi-structured reviews. They used associative rules and manually labelled data for this purpose. Similarly, Shi and MingYu [113] studied a theoretical framework based on product feature mining issues from customer reviews and proposed a DFM (Data, Function, Mining) model for mining product feature structures from such reviews. In another work Zhai et al [128] proposed a semi-supervised method for clustering product features for opinion mining. They used a semi-supervised approach for grouping synonym features.

In contrast we use an approach based on general domain text pre-processing and Latent Dirichlet Allocation [6], which neither relies on the structure of the document nor requires any manually labelled training data for mining latent topics. Therefore, it is applicable to unstructured text and is generalizable to any text collection.

Opinion mining or sentiment classification is a widely studied field. Some of the previous approaches focuses on sentiment-based classification of individual words, phrases, sentences or documents as whole and assume sentiment classification as a binary task (positive or negative) [28, 92, 93, 121].

Eirinaki et al. [32] presented an algorithm for analyzing the overall sentiment of reviews and also identified semantic orientation of the specific



component of the review that leads to specific sentiment. Qui et al. [101] come up with a self-supervised model for sentiment classification. They used a dictionary based approach for sentiment classification. Lin et al. [72] present a weakly supervised sentiment classification approach which is directly incorporated into a topic analysis based on LDA. They proposed a joint sentiment-topic model based on LDA and added an additional sentiment layer between document and topic layers. They incorporated documents' sentiment information as a prior in the model. Joint modelling of topic and sentiments were also addressed by Mei et al. [76]. They proposed a probabilistic Topic-Sentiment Mixture model for mining latent topics and associated sentiments. However, their approach requires a dataset which is already labelled for positive and negative sentiments to learn the sentiment priors for the model. In another work, Ganesan et al. [35] proposed an unsupervised approach for generating ultra-concise aspect related summaries of opinions.

The above mentioned approaches do classify a document as a whole or parts of it as positive or negative and identify the polarity of a text snippet in relation to some aspect. One can use them to classify a document as positive or negative on the basis of positive or negative phrase counts but these approaches do not provide the strength of the sentiment in some numeric form. In another work [92] it is shown that sentiment classification can be generalized into a rating scale. Intuition is one can get better diversification of reviews using a rating scale rather than using binary classification. Kobayashi et al. [61] worked on mining aspect related opinions from the web documents and used domain specific dictionaries of evaluation and aspect phrases, identify candidate aspects and evaluations by dictionary lookup. For sentiment analysis we apply domain independent dictionary based approach [8] which has already been applied successfully in other social web scenarios [83,89]. This approach not only helps in sentiment classification of a document but also provides sentiment scores which can be used to reflect the overall strength of the sentiment.

Result diversification has recently received a lot of attention in the Information Retrieval community. A good overview of the general task of search result diversification and its evaluation in particular is presented in [12]. Agrawal et al. [1] proposed an algorithm based on greedy approach for diversifying search results. In their problem settings, they assumed that each web query has multiple ambiguous intents and these intents can be modelled as topics in an existing taxonomy of information. Both the queries and documents may belong to more than one category of this taxonomy. They also assumed that distribution of intents over categories of taxonomy is already known. They diversified the results with respect to the query intents. The selection of product reviews as a diversification task is addressed by Tsaparas et al. [120]. They formulated the problem as maximum coverage problem and used a greedy approach for solving the diversification task. The

focus, however, is only on a good coverage of product features. In recent work [65] similar to our approach, LDA was used for detecting latent topics in the reviews and star ratings of the reviews as an indicator of sentiment polarity to be used in review diversification. This method relies on star ratings for determining overall sentiment of the review.

However, in our approach we do not need to rely on star ratings for determining sentiment polarity. Furthermore, none of the approaches has addressed the task of diversification of both: feature and sentiment coverage in selected documents.

## 5.6 Summary

In this chapter we looked at the task of selecting a feature-centric sentiment diversified set of end user discussion contributions. The objective of this task is to rank of set of contributions such that the top- $k$  entries cover a wide range of sub-topics or features addressed in a discussion as well as a diversified range of sentiments. We formalized this task as maximum coverage problem and investigated it in the context of product reviews. With the FREuD approach we proposed a solution to this task using a greedy approach. We constructed a real life dataset composed of end user CNET product reviews and developed a gold standard dataset for the purpose of evaluating feature-centric sentiment diversification approaches. The reviews in the gold standard dataset were annotated by human assessors for product features discussed in each review and for sentiment orientation of the reviewer towards the identified features. We evaluated our proposed FREuD approach on this dataset and compared it against two baseline systems. In this empirical evaluation we have been able to show that FREuD significantly outperforms both baseline systems.

## Chapter 6

# Temporal Dynamics of Topics and Authors in Social Media

The world wide web provides a platform for content sharing activities where people can share views, participate in discussions, publish technical domain specific blogs and research papers, thereby, contribute tremendous online contents related to different domains and subject areas. For better understanding of these text contents, they are often categorized with respect to the subject they discuss. These subjects areas are termed as topics. Topics discussed in social media vary with respect to their longevity. Some last for a very brief period and some continue to develop over a period of time and thereby enjoy sustained interest from contributors. It has been observed that topics discussed in collaborative social networks exhibit spikes (sudden topics, linked to current events, or enjoying a limited-time interest) and chatters (more recurring, long term topics) indicating strong correlation [42, 43].

As mentioned in the Scenario 1.3, where a user is interested in tracking a particular topic and finding key authors contributing maximum contents to the topic over a period of time. In such a scenario manual analysis of this tremendous amount of text for finding latent topics, capturing topic evolution, identifying the author's interests and depicting changes in interests is expensive in terms of time and labor. In such situation, the challenge is to provide a model which is able to capture temporal topic dynamics and provides an insight into changing user interests with respect to evolving topics. One such model can be helpful in finding influential authors at different stages of topic evolution and can also be helpful in characterizing authors as pioneers, mainstream or laggards in different subject areas.

We tackle the above mentioned problem using a probabilistic framework which find its roots in Hierarchical Bayesian Statistics and propose Author-Topic-Time (ATT) Model. ATT model extends the well established model

for document collections, the Latent Dirichlet Allocation (LDA) model by augmenting the document contents with authors and time at which the contents are generated. We see authors and timestamp as metadata attached to contents of documents and model the topics, authors and timestamps by a latent multinomial topic distribution and map each entity into common lower dimensional latent topics space. Augmenting documents with timestamps and authors in the ATTs' document generation process helps it in capturing topic dynamics and at the same time author's topical interest, enabling it to find the key authors that are contributing to the specific topics at different stages of topic life cycle.

Therefore, ATT provides a deeper understanding of topical shifts and nature of user collaborations in social environment. We evaluate the efficiency of the model in predicting authors and capturing lifespan of the topics by running the model on subset of abstracts of scientific publications from CiteSeerX dataset and compare the results with the standard LDA. The results obtained can be exploited for social retrieval tasks and recommender systems in recommending specific authors or publications to read for a given user interests or categorizing the authors as pioneers, mainstream or laggards based on their contributions to the topics at different stages of topic life cycle.

The rest of this chapter is organized as follows. Section 6.1 provides an overview of the ATT model which includes model design, description of model parameters and estimation approach used for estimating parameters of the model. In Section 6.2 we list dataset used for evaluating the ATT model with evaluation results and discussion of the results. Section 6.3 provides the related work in topic modeling while results are summarized in Section 6.4.

## **6.1 The Author-Topic-Time Model (ATT)**

### **6.1.1 Model Design**

LDA is a Bayesian multinomial mixture model which has become a state of the art and popular method in text analysis due to its ability to produce interpretable and semantically coherent topics. It uses the Dirichlet distribution to model the distribution of the topics for each document. In LDA, each word is considered sampled from a multinomial distribution over words specific to this topic. LDA is a well-defined generative model and generalizes easily to new documents without overfitting. Since LDA is highly modular and hierarchical, therefore, it can easily be extended. Many extensions to basic LDA model have been proposed to incorporate document metadata. The simplest method of incorporating the metadata in generative topic models is to generate both the words and the metadata simultaneously given hidden topic variables. In this type of model, each topic has a distri-

bution over words as in the standard model, as well as a distribution over metadata values.

We extend LDA by incorporating authors and timestamps of the documents into the Author-Topic-Time (ATT) model. It is common to represent probabilistic graphical models as a set of random variables and their conditional dependencies using Bayesian Networks. Figure 6.1 shows a graphical representation of ATT model and Table 6.1 represents different notations used in the modeling process of ATT. Each relation defines a node in the model in terms of other nodes which are referred as parent nodes. The parent and child nodes taken together forms a directed acyclic graph. The top-level nodes with no parents in the graph are constants. The relationship between nodes can be either of stochastic relation defining random variable in the model graph or deterministic relations representing deterministic node. The values of deterministic nodes are computed from the values of parents nodes. The latent random variables of interest are normally non shaded circles in the diagram and are unobserved depicting the model parameters, whereas the observed variables are shaded circles and the directed arrows show their conditional dependencies.

In the modeling process, we assume that authors are interested in writing about more than one topic, thus each author is modeled as having multinomial distribution over topics. We further assume that each document addresses more than one topic, thus each document is modeled as having multinomial distribution over topics. ATT is a generative model of documents labeled with timestamps and authors. In the first step of this process we specify a language model that depicts the document generation in the real world by assuming specific parametrized distributions without the data being observed. In the second step after the data has been observed, we reverse the process in step one and use statistical inference techniques to find which topic model is most likely to have generated the data. This involves estimating the values of distribution parameters that can best explain the observed data.

The document generation process of ATT starts by picking each of the  $N_d$  words in the document  $d$ . Then we sample an author  $a$  uniformly at random from the list of authors  $A_d$  for document  $d$ . Then a topic  $z$  is chosen randomly from author specific distribution of topics  $\theta_a$ . After selecting a topic  $z$ , a word  $w$  is sampled from the topic specific distribution over words  $\phi_z$ . At the same time a timestamp  $t$  is generated from topic specific beta distribution  $\psi_z$ . Typically every document has one timestamp associated with it, therefore, in the generative process of ATT each word assumes the same timestamp as of enclosing document during training step.

The generative process of the ATT model which corresponds to the process used in Gibbs sampling for parameter estimation is described as follows.

1. For each author  $a = 1 \dots |A|$ , draw  $\theta_a \sim Dir(\alpha)$

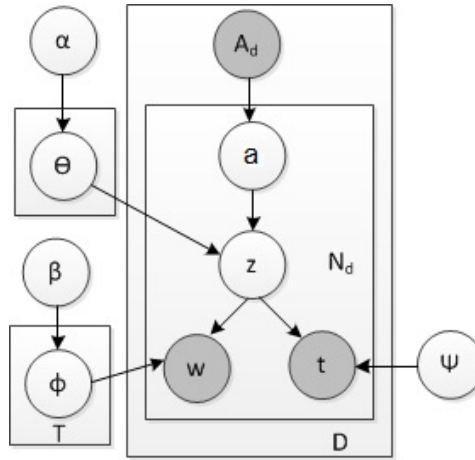


Figure 6.1: Document generation process as specified by the ATT model.

Table 6.1: Notation used in the modeling process

Log-odds	Tweet
$T$	number of topics
$D$	number of documents
$A$	number of authors
$N_d$	number of word tokens in document $d$
$\theta_a$	the multinomial distribution of topics specific to the author $a$
$\phi_z$	the multinomial distribution of words specific to topic $z$
$\psi_z$	the beta distribution of time specific to topic $z$
$z_{di}$	the topic associated with the $i$ th token in the document $d$
$w_{di}$	the $i$ th token in the document $d$
$t_{di}$	the timestamp associated with the $i$ th token in the document $d$
$a_{di}$	the author associated with the $i$ th token in the document $d$
$\alpha, \beta$	Dirichlet priors

2. For each topic  $t = 1 \dots |T|$ , draw  $\psi_z \sim Dir(\beta)$
3. For each document  $d$ , pick an author  $a$  from the list of authors  $A_d$  and draw a multinomial  $\theta_a$  from Dirichlet prior  $\alpha$ ; then for each of the  $N_d$  words,  $w_i$ ,
  - Draw a topic  $z_{d_i}$  from multinomial  $\theta_a$ ;

- Draw a word  $w_{d_i}$  from multinomial  $\phi_{z_{d_i}}$ ;
- Draw a timestamp  $t_{d_i}$  from Beta  $\psi_{z_{d_i}}$

Where  $Dir(\alpha)$  indicates the Dirichlet distribution [64] and in Bayesian statistics is often associated to multinomial data sets for the prior distribution of the probability parameters. It is a continuous multivariate probability distribution having a vector  $\alpha$  of parameters which are strictly positive numbers. The probability density function for Dirichlet distribution of order  $k \geq 2$  with parameters  $(\alpha_1, \dots, \alpha_k)$  is given by

$$p(t_1, \dots, t_k | \alpha_1, \dots, \alpha_k) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^{k-1} \Gamma(\alpha_i)} \left[ \prod_{i=1}^{k-1} t_i^{\alpha_i-1} \right] \left[ 1 - \sum_{i=1}^{k-1} t_i \right]^{\alpha_k-1} \quad (6.1)$$

Where  $\Gamma(\alpha_i)$  is a gamma function. Due to functional relationship between  $k$  variables (summation to one), their joint probability distribution is degenerated. This is why the density is proposed on the first  $k-1$  variables, the last one being given as  $t_k = 1 - \sum_{i=1}^{k-1} t_i$ . When all  $\alpha_i = 1$ , the Dirichlet distribution reduces to the uniform distribution. If  $k = 2$ , it is easy to see that  $t_1 \sim Beta(\alpha_1, \alpha_2)$ ,  $t_2 \sim Beta(\alpha_2, \alpha_1)$  and  $t_1 + t_2 = 1$ , this is why Dirichlet distribution is considered as generalization of the the Beta distribution. Beta distribution is a continuous probability distribution defined on the interval  $[0, 1]$ . It describes a family of curves that are unique in that they are nonzero only on the interval  $[0, 1]$ . It has two free shape parameters labelled as  $\alpha$  and  $\beta$  that are exponents of the random variable. In Bayesian analysis, Beta distribution is used as a prior distribution for binomial proportions. The probability density function of Beta distribution is given by

$$p(x | \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} I_{(0,1)}(x) \quad (6.2)$$

Where  $I_{(0,1)}(x)$  is an indicator function which ensures that only values of  $x$  in the range  $[0, 1]$  have nonzero probability,  $B(\alpha, \beta)$  is the beta function and is defined as

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} \quad (6.3)$$

If  $\alpha = \beta = 1$ , the Beta distribution reduces to the uniform distribution over  $[0, 1]$ . Different values of  $\alpha$  and  $\beta$  give rise to different shaped curves.

The ATT model has three sets of unknown parameters; the author distribution over topics  $\theta$ , the topic distribution over words  $\phi$  and the topic distribution over time  $\psi$ . Both  $\theta$  and  $\phi$  have multinomial distributions with symmetric Dirichlet priors having the hyperparameters  $\alpha$  and  $\beta$  respectively. To avoid time discretization we use a continuous per-topic parametric Beta

distribution  $\psi$  as used in [124] over absolute time values in the generative process, which gives a natural distribution of topics over time. Beta distribution is family of continuous probability distributions having two free parameters termed as shape parameters. We normalize the time-stamps of the documents to values between 0 and 1 for parameter estimation. One advantage of using continuous Beta distribution over other distributions is that it fits better to the temporal part if the data is sparse. The disadvantage of Beta distribution is its inability to capture multiple spikes in a topic life cycle. This disadvantage is out-weighed by its ability to fit the sparse data which otherwise can lead to two different topics if there is a big time gap in a topic life cycle.

### 6.1.2 Model Parameters

One of the model parameters which needs to be fixed before run is decision upon the the number of topics for the corpus. One can choose the number of topics either manually by running the model for different number of topics and then subjectively judging which number produces more cohesive and distinctive topics, or by using the parametric and non parametric methods. Parametric method is to plot model log-likelihood against the number of topics and select the number of topics for which the log-likelihood reaches the maximum. This approach over-fits the training data as for different enough document-term distributions, number of topics equal to number of documents would give the maximum log-likelihood. The other way is to use log-likelihood of held-out data to avoid over-fitting of the training data, but [15] pointed that topic models which perform better on held-out likelihood may infer less semantically meaningful topics and demonstrated that traditional metrics of model selection do not capture whether topics are coherent or not.

As we need refined topics in order to be useful, therefore, we use manual approach and run the model for different number of topics and qualitatively judged the topics and select the number of topics for the corpus which produce semantically meaningful and coherent topics. We found from different runs of the model with varying number of topics that the model produced better topics when run with 100 topics on the CiteSeerX dataset and therefore set the number of topics to  $K=100$  and fix the hyper-parameters  $\alpha = 50/K$  and  $\beta = 0.01$  accordingly.

### 6.1.3 Parameters Estimation

We have already mentioned in Section 6.1.1 that we model authors and topics using multinomial distribution and Beta distribution for modeling the time. The multinomial distribution is a generalization of binomial distribution. Binomial distribution is used to model events where exactly one of



the two outcomes are possible in a number of  $n$  independent Bernoulli trials with a fixed success probability  $p$  for each trial. In multinomial distribution the Bernoulli distribution is replaced with categorical distribution where the trial results in one of the possible  $k$  finite fixed outcomes (K-dimensional Bernoulli) with probabilities  $p_1, \dots, p_k$ . In the context of topic modeling, this  $k$  corresponds to number of topics in the model. Once we have fixed the model distributions and data has been observed, in the next step we need to estimate distribution parameters which are most likely to have generated the observed data and compute the probability of new observation  $\tilde{x}$  given previous observations.

We take Bayesian approach for estimating the parameters of ATT. Exact inference of the parameters of LDA type models is intractable, therefore, we need to use approximate inference algorithms such as mean-field variational expectation maximization [6], expectation propagation [80], Gibbs sampling [39, 40] etc. For ATT we use Gibbs sampling to perform approximate inference because its relatively simple algorithm for approximate inference. In the ATT model three parameters  $\theta, \phi, \psi$  are estimated. The probability of the corpus  $w$  in ATT conditioned on  $\theta, \phi$  and  $\psi$  is

$$p(w|\theta, \phi, \psi, \mathcal{A}) = \prod_{d=1}^D p(w_d|\theta, \phi, \psi, a_d) \quad (6.4)$$

In the ATT model there are three latent variables  $z, a$  and  $t$ . Each set  $(z_i, a_i, t_i)$  of these latent variables is drawn as block conditioned on all other variables. We begin with the joint probability of dataset, and using the chain rule we obtain conditional probability for

$$p(z_i = j|w_i = m, z_{-i}, x_{-i}, t_{-i}, w_{-i}, a_d) \quad (6.5)$$

where  $z_i, x_i, t_i$  represent topic, author and time assigned to  $w_i$  whereas  $z_{-i}, x_{-i}, t_{-i}$  are all other assignments of that topic, author and time excluding the current assignment.  $w_{-i}$  represents all other words in the document set and  $a_d$  is the observed author of the document. Learning joint probabilities of these three latent variables enables us to query the model conditioned on any combination of these variables using Baye's rule. For example given the author and time find the authors interest in that time period  $p(\phi_d|a, t)$  or given the topic and time find the top authors contributing to the topic in that time  $p(\theta_d|z, t)$ .

We used JAGS's (Just Another Gibbs Sampler) [98] implementation of Gibbs sampler for estimating the ATT parameters. JAGS is a bundle software that provides routines for analysis and inference on Bayesian graphical models using Markov Chain Monte Carlo (MCMC) simulation. There is no graphical user interface provided in JAGS for model building, but one can describe the model using BUGS language.

JAGS can be used from command prompt using specified commands for the given task or using script file containing those commands. We used command line interface of JAGS for specifying the model, data and subsequent computing of the model parameters. In JAGS, there is five step process for generating samples from the posterior distribution of the model parameters. These five steps are:

- Model definition, which includes the definition of model and definition of data using BUGS language. The script which is used in JAGS to describe ATT and is equal to the graphical model of ATT (Figure 6.1) is as follows:

```

model{
  for( k in 1 : K ){
    phi[k , 1:V]~ddirch(beta[])
  }
  for ( ii in 1 : A ) {
    theta[ii,1:K]~ddirch(alpha[])
  }
  for( j in 1 : K ){
    alphab[j]~dunif (1,10)
    betab[j]~dunif (1,10)
  }
  for( m in 1:M ){
    for( n in 1:wdim[m] ){
      x[m,n]~dcat(a[m*A-A+1:m*A])
      z[m,n]~dcat(theta[x[m,n],1:K])
      w[wstart[m]+n-1]~dcat(phi[z[m,n] , 1:V])
      t[wstart[m]+n-1]~dbeta(alphab[z[m,n]],betab[z[m,n]])
    }
  }
}

```

- Compilation of the model includes generating a graph of the model in the system memory.
- Model initialization includes setting values of model parameters and sampling of parameter from prior distribution.
- Adapting and burn-in: As the number of iterations of the sample increases the sampler starts converging to the target distribution means posterior distribution of the parameters. An initial burn-in period (iterations) are discarded to overcome the bias resulting from the prior distribution of the parameters. Later iterations are then used to observe for convergence towards target distribution.

- Monitoring includes recording of the sampled values of nodes at each iteration.

Once the desired number of iterations of the Gibbs sampler is complete, one can dump the JGAS output into R compatible files for subsequent analysis. The R language provide several packages for analyzing the JAGS output which can be used form R environment.

As, JAGS provide highly generic implementation of Gibbs sampler, therefore it is quite expensive in terms of time and memory for complex models to achieve the convergence. For our model and dataset it took about 40 Gb of RAM and 3 weeks of time to finish the 2000 iteration of Gibbs sampler. Almost one third of this time was spent in model compilation and initialization, while rest was taken by the burn-in and adaptation of the model. The results presented in this chapter are generated from the analysis of such files output by JAGS.

## 6.2 Experiments and Evaluation

### 6.2.1 Dataset

To show the effectiveness of our approach in capturing the topic evolution and finding the main contributors for different topics, we run the model on subset of abstracts from CiteSeerX<sup>1</sup> publications. The dataset consists of abstracts and titles of research papers published in computer science domain from 2001 to 2009. We selected 18 authors from the crawl with each author having more than 150 publications in the above mentioned time period. We selected authors for which we are able to find their profiles in the web to manually check the results of our model with the authors' interests as reflected from their publications available on the web. The minimum limit of 150 publications is applied to overcome the sparsity in data and to have sufficient text for capturing authors interest over time. The dataset is divided into test set and training set. Test set contains 3230 documents and training set contains 800 documents. We preprocessed the data to remove stop words and noise by removing highly frequent terms and terms occurring in less than 10 documents. We used Porter stemmer [100] to reduce the word inflection to their stems.

### 6.2.2 Evaluation Method

Evaluation of probabilistic topic models poses a certain challenge because of their unsupervised nature makes model selection a difficult task. Generally there are two ways to evaluate topic models. One to evaluate the model in its application scenario such as document classification or in an information

---

<sup>1</sup><http://citeseerx.ist.psu.edu>

retrieval task. Second to evaluate the model that how good it generalizes or fits to unseen documents or data given a training collection. In the second case a better model will give higher likelihood to held-out data. This type of evaluation of probabilistic models is done by using perplexity which is a standard measure for estimating the performance of probabilistic models. Perplexity is defined as the ability of the model to predict words to new documents. It gives a measure of how much the model is surprised when it sees data which is unseen previously. It is defined as the inverse of geometric mean of per-word likelihood of held-out data and is given below as defined in [46].

$$perplexity(\tilde{\mathcal{W}}) = \exp \left\{ -\frac{\sum_{d=1}^M \log p(\tilde{\mathcal{W}}_d)}{\sum_{d=1}^M N_d} \right\} \quad (6.6)$$

where  $\tilde{\mathcal{W}}_d$  is the word vector for document  $d$  in held-out data.

The qualitative evaluation is generally done by looking at the top terms in each topic for their semantic cohesiveness (semantically fit together and convey theme of the topic) and by showing that the topics discovered by the model are distinct when compared with the base model. A better model will produce more distinct topics that are semantically coherent. In this case we use KullbackLeibler divergence (KL-divergence) [66, 67], which is used to measure difference between two non-symmetric probability distributions. The details of these measures are given in Chapter 3.

### 6.2.3 Evaluation Results

We compare the generalization performance of ATT with LDA. We randomly split our dataset into training and test set and used 80% of the dataset for training while keeping 20% as held-out data for test. The perplexity scores for both models are given in figure6.2.

These results indicate that ATT better generalize to the unseen document as compared to the baseline LDA model. The improvement in generalization performance of ATT can be explained by its ability to better model document-specific topic distributions and the topics detected by LDA are more heterogeneous than detected by LDA. If a word which has small probability in the topics of training document, then it will cause an increase in perplexity. As the number of topics  $k$  increase the resultant topics get more specific and the probabilities assigned to words get smaller in each topic. This reduces the chances that the training documents' topic proportions will cover all the words in unseen document, therefore perplexity increases.

We visualize each topic by showing the top K terms in descending order of the probability values assigned to terms as being the most representative terms of that topic. Table 6.2 shows topic terms from 9 topics selected at random from 100 topics discovered by the ATT from the CiteSeerX dataset.

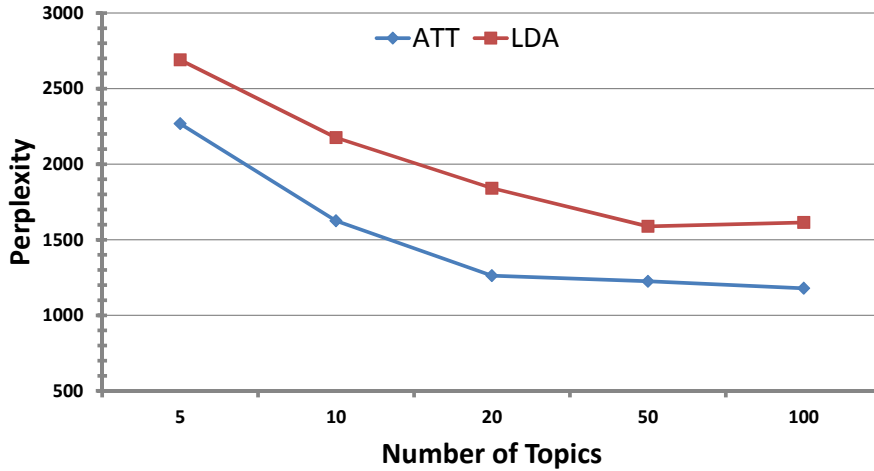


Figure 6.2: Plot showing average word perplexity scores achieved by ATT and LDA model. Lower perplexity scores indicate better generalization performance.

To verify subjectively that top terms in the topics produced by the ATT model are semantically more cohesive (terms fit together to convey the theme of topic) than top terms of the topics produced by our base model (LDA), we also show 9 similar topics in Table 6.3 that are produced by the LDA from the same dataset. The similarity between the topics of ATT and LDA is computed using KL-divergence. These results are obtained by sampling from 2000<sub>th</sub> iteration of Gibbs sampler.

From topic visualizations in Table 6.2 and in Table 6.3, we subjectively see that top  $K$  terms that are assigned high probability in topics produced by ATT are better semantically related to each other than the terms in topics captured by LDA. Further, we see that the average KL-divergence as shown in Table 6.4 between topics produced by ATT is higher than the average KL-divergence between topics produced by LDA indicating that topics produced by ATT are more distinct than topics produced by LDA confirming our visual observation of the topics.

Table 6.5 presents symmetric KL divergence of 4 sample topics from ATT for which we also show the topic life cycle and top authors in Tables 6.6 to 6.9 for each topic. High KL divergence values show that topics produced by ATT are distinct to each other and ATT is able to capture the distinct topics.

Tables 6.6 to 6.9 shows the top 5 terms and the top 4 authors for each topic and respective beta distribution capturing the topic activity. The interesting observation from qualitative analysis of the results is that the activities in the Semantic Web and Database System topics are correlated. As one topic starts gaining, the activity in other topic starts decreasing. Top

Table 6.2: Representation of 9 topics from a 100-topic run of Gibbs Sampler for CiteSeer dataset discovered by ATT model

Topic 9		Topic 26		Topic 21	
Word	Prob.	Word	Prob.	Word	Prob.
storag	0.027969	ontolog	0.04307	protocol	0.039171
disk	0.023475	web	0.037838	control	0.020078
failur	0.018981	semant	0.028983	rout	0.019588
reliabl	0.017483	languag	0.022543	packet	0.018609
select	0.015985	rdf	0.018518	wireless	0.01763
server	0.014986	knowledg	0.015298	access	0.016651
fault	0.014986	schema	0.014896	servic	0.014692
cach	0.012988	servic	0.012481	util	0.014692
Topic 79		Topic 93		Topic 63	
Word	Prob.	Word	Prob.	Word	Prob.
file	0.028499	sensor	0.026113	peer	0.018658
metadata	0.022959	channel	0.025711	control	0.015105
secur	0.019793	access	0.023703	cach	0.013329
analysi	0.019002	protocol	0.020088	resourc	0.010664
safeti	0.015045	schedul	0.016874	manag	0.01022
share	0.014253	optim	0.013661	search	0.009775
storag	0.012671	wireless	0.012858	server	0.009775
express	0.012671	resourc	0.012054	dynam	0.009331
Topic 84		Topic 90		Topic 62	
Word	Prob.	Word	Prob.	Word	Prob.
access	0.024221	cluster	0.021389	delay	0.020169
traffic	0.022284	gene	0.017015	circuit	0.01274
sensor	0.018411	transact	0.015557	pair	0.011679
rang	0.017442	array	0.012155	optim	0.010618
load	0.017442	estim	0.011669	gate	0.010087
composit	0.0126	studi	0.011183	fault	0.010087
dynam	0.011631	construct	0.010697	function	0.009026
bank	0.011631	express	0.010697	axiom	0.009026

authors for “*SemanticWeb<sub>6.8</sub>*” and “*DatabaseSystems<sub>6.9</sub>*” topics are well known authors in this field in our dataset. Results also shows that as the topic of semantic web started to emerge, influential authors in the database systems topic shifted to semantic web topic.

Authors that are assigned high probability for a topic when it starts emerging can be seen as “topic pioneers” who conduct innovative research in that topic. Moreover, active authors that frequently change their topics

Table 6.3: Representation of 9 topics from CiteSeer dataset discovered by LDA

Topic 3		Topic 4		Topic 9	
Word	Prob.	Word	Prob.	Word	Prob.
node	0.013466	peer	0.019961	ontolog	0.024649
optim	0.012448	queri	0.015684	logic	0.021779
cluster	0.012109	databas	0.014258	role	0.017559
test	0.011543	metadata	0.011407	knowledg	0.017052
graph	0.010298	resourc	0.008944	languag	0.015871
rout	0.008827	view	0.008944	descript	0.015026
structur	0.008148	grid	0.008426	web	0.015026
point	0.008035	search	0.008037	reason	0.014351
Topic 11		Topic 6		Topic 19	
Word	Prob.	Word	Prob.	Word	Prob.
ontolog	0.021625	queri	0.015773	node	0.015152
learn	0.013778	build	0.013414	traffic	0.013258
logic	0.010639	optim	0.011498	peer	0.013123
delay	0.009767	function	0.010909	optim	0.009606
tree	0.009593	size	0.010909	imag	0.00947
reason	0.009418	databas	0.008845	rout	0.009335
function	0.009244	oper	0.008403	watermark	0.008659
power	0.008895	logic	0.007813	symbol	0.008253
Topic 54		Topic 85		Topic 69	
Word	Prob.	Word	Prob.	Word	Prob.
ontolog	0.021625	queri	0.015773	node	0.015152
learn	0.013778	build	0.013414	traffic	0.013258
logic	0.010639	optim	0.011498	peer	0.013123
delay	0.009767	function	0.010909	optim	0.009606
tree	0.009593	size	0.010909	imag	0.00947
reason	0.009418	databas	0.008845	rout	0.009335
function	0.009244	oper	0.008403	watermark	0.008659
power	0.008895	logic	0.007813	symbol	0.008253

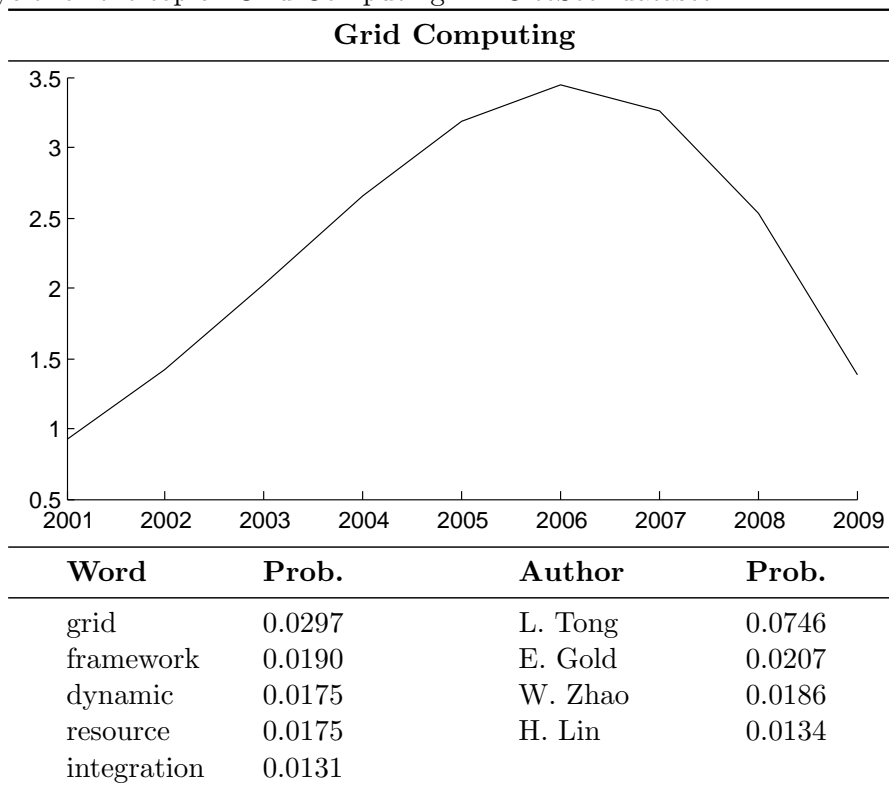
Table 6.4: Average KL divergence between topics for ATT and LDA

Model	Average KL Divergence
ATT	14.5934
LDA	8.4524

Table 6.5: Symmetric KL divergence for pairs of topics shown in Table 6.6, 6.7, 6.8, 6.9

Topic Pair	KL Divergence
Image Analysis - Grid Computing	15.4372
Grid Computing - Semantic Web	14.942225
Semantic Web - Database Systems	14.4469
Database Systems - Image Analysis	14.000675

Table 6.6: Top terms, influential authors and beta PDF depicting topic life cycle for the topic “Grid Computing“ in CiteSeer dataset

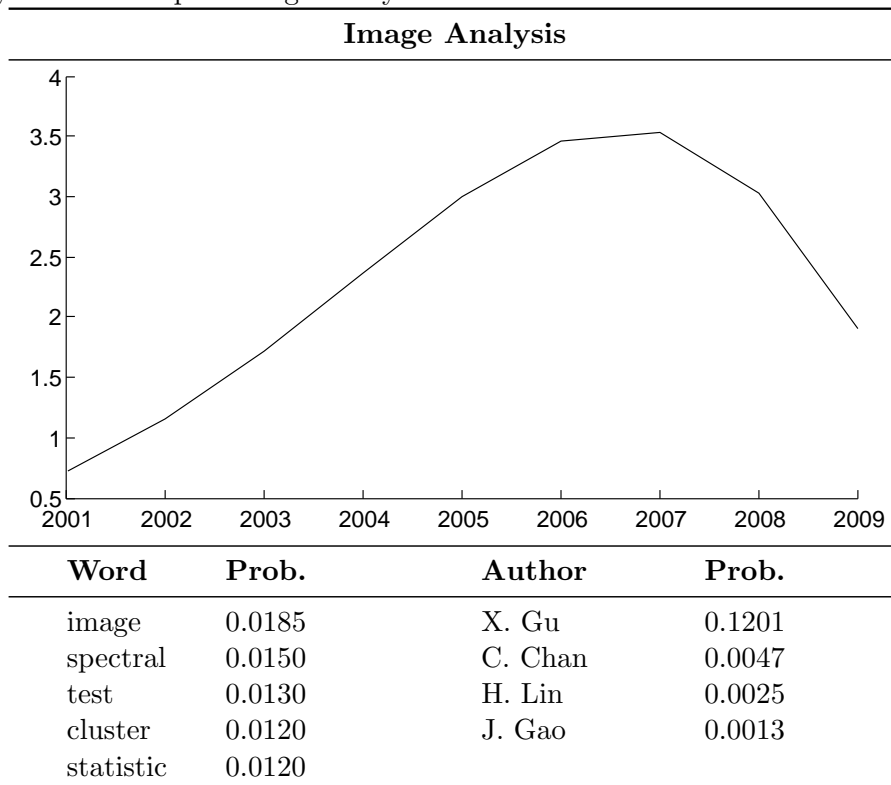


of interest can be considered as “trend setters” in the respective research community. On the other hand, authors that have high probability at the peak topic activity can be seen as “mainstream” researchers that follow general trends and interests of the community. Finally, authors that have time-independent profiles with stable topics of interest can be recognized as foundational researchers that act independently of fluctuating trends and popular issues.

Table 6.10 shows author pairs and their symmetric KL divergence when



Table 6.7: Top terms, influential authors and beta PDF depicting topic life cycle for the topic “Image Analysis“ in CiteSeer dataset

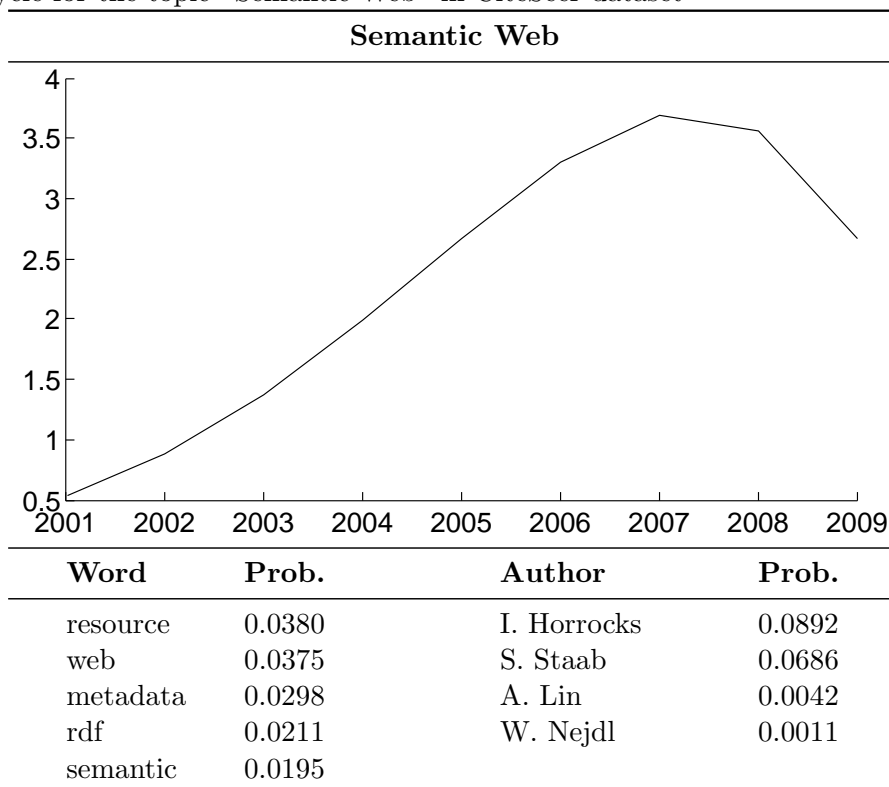


both author are present in top K authors of the same topic. The subscript numbers with author names indicate topics for which comparisons are made and are taken from topics shown in Tables 6.6 to 6.9. Small values of KL divergence show that both author share similar interests. To mention S. Staab and I. Horrocks are well known authors in Semantic Web area and therefore share similar interest as shown in our results also. Table 6.11 shows author pairs and their symmetric KL divergence when one author has high probability assigned in one topic and the other has high probability assigned in another topic. The large KL divergence values show that both authors have dissimilar interests suggesting that ATT is able to capture these dissimilarities.

#### 6.2.4 Application Scenarios

Prediction power of the ATT model can be used in variety of ways. One such scenario is to recommend target authors whose research paper to read given user's interest at a given time point. That is, given some terms which describes an authors' interest the task is to generate a ranked list of target

Table 6.8: Top terms, influential authors and beta PDF depicting topic life cycle for the topic ‘‘Semantic Web’’ in CiteSeer dataset



authors whose interest are highly likely to be similar with the user interest. This is achieved by computing the topic assignments to given user using the query terms from the posterior distributions of trained model. Then highly likely similar authors are found by computing the similarity between user and existing authors topic distributions in the model using KL divergence as distance measure. The small values of KL divergence between a pair of authors means both authors are similar. The target authors are then ranked based on the values of KL divergence between the user and target authors.

We can also use ATT for classification of authors being as Pioneers, main stream or laggards. This can be achieved by looking at the topic life cycle and finding the time when it starts emerging and then conditioning the model on the time and topic to find the authors that have high probability for the topic at that time. These top authors returned by the model are the pioneers in that topic. Similarly, conditioning the model on the time when it has peak activity for given topic will help to find authors that are mainstream authors for that topic.

Table 6.9: Top terms, influential authors and beta PDF depicting topic life cycle for the topic “Database Systems“ in CiteSeer dataset

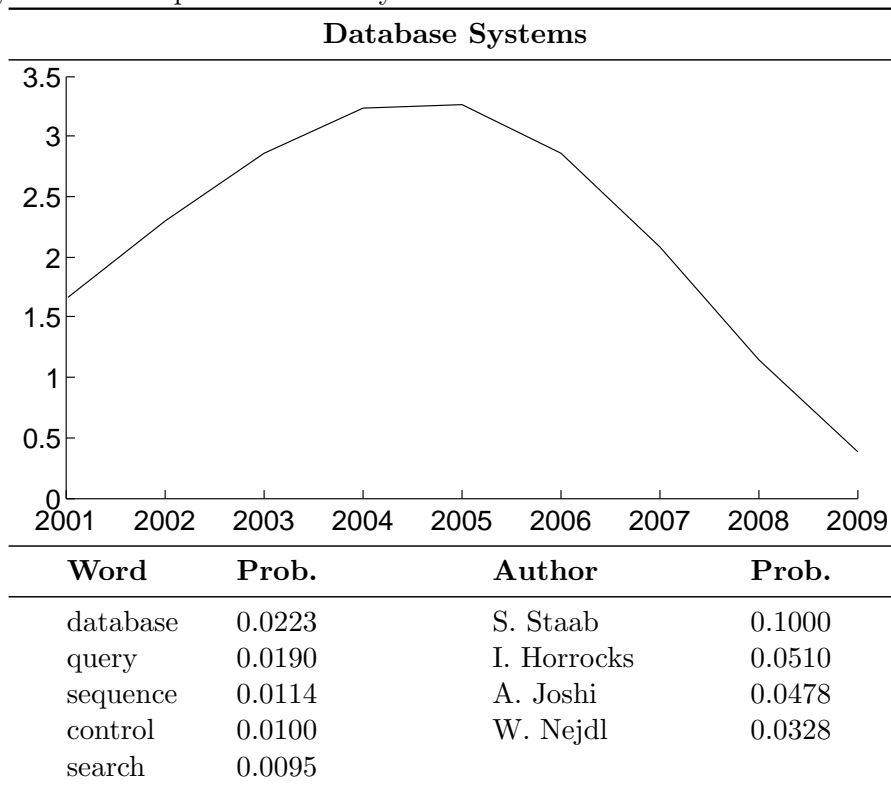


Table 6.10: Intra-topic symmetric KL divergence for different pairs of authors

Author Pair	KL Divergence
S. Staab <sup>a</sup> – W. Nejdl <sup>a</sup>	1.82
S. Staab <sup>b</sup> – I. Horrocks <sup>b</sup>	2.40
C. Chan <sup>c</sup> – H. Lin <sup>c</sup>	2.19
H. Lin <sup>c</sup> – J. Gao <sup>c</sup>	2.06

<sup>a</sup> Database Systems 6.9<sup>b</sup> Semantic Web 6.8<sup>c</sup> Image Analysis 6.7

## 6.3 Related Work

### 6.3.1 Topic Modeling

In probabilistic topic modeling a “*Topic*” is seen as a multinomial distribution over a vocabulary that assigns high probability to a set of words that tend to appear in the similar documents. A qualitatively “better topic” is

Table 6.11: Inter-topic symmetric KL divergence for different pairs of authors

Author Pair	KL Divergence
S. Staab <sup>a</sup> – L. Tong <sup>d</sup>	8.87
I. Horrocks <sup>a</sup> – X. Gu <sup>c</sup>	8.69
S. Staab <sup>b</sup> – X. Gu <sup>c</sup>	8.64
L. Tong <sup>d</sup> – X. Gu <sup>c</sup>	7.64

<sup>a</sup> Database Systems 6.9

<sup>b</sup> Semantic Web 6.8

<sup>c</sup> Image Analysis 6.7

<sup>d</sup> Grid Computing 6.6

that in which words that have high probability are semantically related to each other and a human subject is able to say that “these words are about X”, where X can be any domain for example, business, computer science, chemistry etc. There is no consensus in literature on what could be a formal definition of a topic model. So, we see a “*Topic Model*” as a model of the generative process by which documents are created and captures the word co-occurrence patterns in a document corpus to produce semantically coherent topics.

### 6.3.2 Probabilistic Topic Models

A variety of statistical models have been proposed for topic-based analysis and modeling of text documents. To name few of them are unigram model, mixture of unigram model [90], latent semantic analysis(pLSA) [48] and Latent Dirichlet Allocation (LDA) [6].

Under the unigram language model, the words of every document are drawn from a multinomial distribution  $\theta$ . The unigram model uses strong independence assumption that words are drawn independently from a multinomial distribution and throws away all conditioning context, and estimates each term independently. Which is,

$$p(w_{1:n}) = \prod_{i=1}^n p(w_i|\theta) \quad (6.7)$$

It is argued that each document in the documents corpus has a distinct topic and [90] has developed mixture of unigram model based on the unigram model. Under this model the document generative process corresponds to the following steps:

1. For each word in the document
  - Draw a topic  $z \sim \theta_z$

- Draw the word from topic specific distribution  $w \sim \theta_z$

The document probability is given by,

$$p(w) = \sum_z p(z) \prod_{n=1}^N p(w_n|z) \quad (6.8)$$

The assumption in mixture model that each document is generated by one topic is relaxed by probabilistic latent semantic analysis (pLSA) [48]. In pLSI each document is generated by the activation of multiple topics, and each topic is modeled as multinomial distributions over words and is given by

$$p(w, d) = p(d) \sum_z p(w_n|z) p(z|d) \quad (6.9)$$

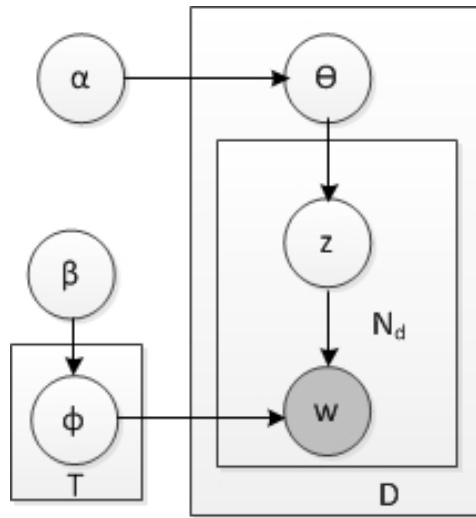
However, pLSA model uses a distribution indexed by training documents, which means the number of parameters being estimated in pLSA grow linearly with the number of training documents. The parameters for a k-topic pLSA model are k multinomial distributions of size V and M mixtures over the k hidden topics. This gives kV + kM parameters and therefore linear growth in M. The linear growth in parameters suggests that the model is prone to over-fitting in many practical applications.

Latent Dirichlet Allocation (LDA) [6] overcomes the problems of pLSA by using the Dirichlet distribution to model the distribution of the topics for each document.

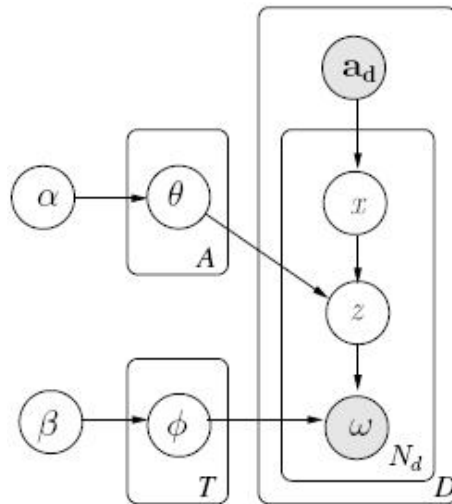
LDA (Figure 6.3(a)) is a Bayesian network that generates a document using a mixture of topics. In its generative process, for each document  $d$ , a multinomial distribution  $\theta$  over topics is randomly sampled from a Dirichlet distribution with parameter  $\alpha$ , and then to generate each word, a topic  $z$  is chosen from this topic distribution, and a word,  $w$ , is generated by randomly sampling from a topic-specific multinomial distribution  $\phi_z$ . The robustness of the model is greatly enhanced by integrating out uncertainty about the per-document topic distribution  $\theta$ .

Since LDA is a hierarchical model, it is easy to extend and include additional parameters of interests. There exist many models which extend LDA and are used for variety of purposes. Examples of such model includes, Topics over Time model [124] of Wang and McCallum, Continuous Time Dynamic Topic Models [123] of Wand and Blei, the Group-Topic model of Wang, Mohanty and McCallum [125], Author-Topic model [105] of Rosen-Zvi, Griffiths, Steyvers and Smyth, Linked Topic and Interest Model [20] of Cheng and Li. The approaches that are directly related to our model includes Author-Topic model and Topic over Time model and are discussed below.

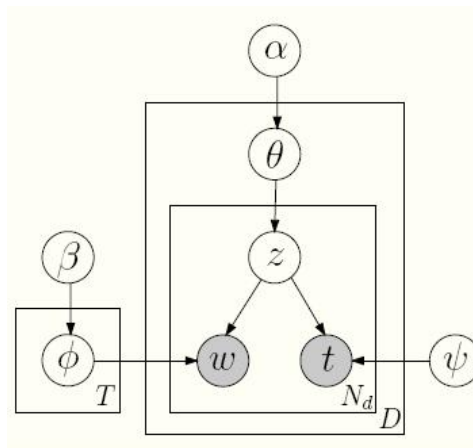
The Author-Topic model [105] is a similar Bayesian network (Figure 6.3(b)), in which each author's interests are modeled with a mixture of topics. In its



(a) Topic-Word (LDA)



(b) Author-Topic



(c) Topics over Time (TOT)

Figure 6.3: Three related Bayesian network models for document generation.

generative process for each document  $d$ , a set of authors,  $a_d$ , is observed. To generate each word, an author  $x$  is chosen uniformly from this set, then a topic  $z$  is selected from a topic distribution  $\theta_x$  that is specific to the author, and then a word  $w$  is generated from a topic-specific multinomial distribution  $\phi_z$ .

The Topics over Time (TOT) [124], a topic model that explicitly models time jointly with word co-occurrence pattern (Figure 6.3(c)). TOT parametrizes a continuous distribution over time associated with each topic, and topics are responsible for generating both observed timestamps as well as words. Parameter estimation is thus driven to discover topics that simultaneously capture word co-occurrence and locality of those patterns in time.

### 6.3.3 Parameter Estimation

Different approaches has been used in the topic-based probabilistic models for parameter estimation. These approaches includes Maximum likelihood estimation (MLE), Maximum a posteriori estimation (MAP) and Bayesian estimation. Expectation-maximization (EM) [47] is used to find the direct estimates of model parameters for MLE and MAP approaches. While variational EM [6], expectation propagation [29], Gibbs Sampling [41] algorithms provide approximate inference of the model parameters in Bayesian estimation. Blei [6] suggested to use approximate methods where parameters  $\theta$  and  $\phi$  can be integrated out because explicit estimate methods suffer from problem of local maxima in topic models. In our experiments, we use Gibbs sampling for approximate inference because it is relatively a simple method for estimating parameters in high-dimensional models.

None of the given approaches models documents and author together with the temporal information. In this contribution, we propose ATT: a model of topic dynamics in social media which connect the temporal topic dependency with the social actors, thereby, providing an insight into the evolution of topics over time along with capturing the author interests for a given time period.

## 6.4 Summary

In this chapter we have proposed a probabilistic approach that models text, authors and timestamps in a given set of documents thus enabling us to capture temporal topic activity and finding out influential authors for the captured topics. Joint modeling and learning posterior probabilities of text, author and time allows us to query model for any arbitrary combination of these variables conditioned on each other for finding information about how author's interests change over time and how activity in topics changes with emergence of new topics.

Results from the application of this model to the CiteSeer dataset show the applicability of the model to arbitrary document collections with author and temporal information for detecting topics trends, topic evolution and author's interests.

In theory, the presented approach can be used for variety of applications. For example authors that are assigned high probability for a topic when it starts emerging can be seen as "topic pioneers" who conduct innovative research in that topic. Moreover, active authors that frequently change their topics of interest can be considered as "trend setters" in the respective research community. On the other hand, authors that have high probability at the peak topic activity can be seen as "mainstream" researchers that follow general trends and interests of the community. Finally, authors that have time-independent profiles with stable topics of interest can be recognized as foundational researchers that act independently of fluctuating trends and popular issues. From the application perspective, this knowledge can be exploited in a variety of ways, e.g. for advanced impact ranking, similarity-based contact recommendation for future collaborations, or better summarization of recent research trends and prediction of their further evolution.

However, for work presented in this thesis we do not run above mentioned queries due to the time and space required for completing this kind of analysis. Like most three dimensional models, ATT also suffers from scalability issues. The current implementation of ATT does not scale well with the large text collections. The scalability issue is also coupled with the way JAGS is currently implemented. The current implementation of JAGS uses old FORTRAN math libraries which are slow and do not scale to large data sets. Another disadvantage of the model is its inability to capture multiple spikes in the topic life cycle. This problem is due to the use of Beta distribution which can not capture multiple peaks in the topic life cycle. In future multimodal distribution may be tried to get around this problem and capture multiple peaks of the topic life cycle.



## Chapter 7

# Conclusions

The research reported in this thesis address automatic analysis of the social media contents. We have looked into different social media platforms that provide opportunities to its users for sharing content with others and are, thus, viewed as a potential source of high quality information on various topics. The research questions we have addressed in this thesis, stem from three example scenarios taken one from each: Twitter, the CNET product review portal and the CiteSeerX. In example scenarios, we have looked at user information needs that are specific to these social media platforms and have proposed methods to overcome the problems faced when retrieving relevant information. Our first contribution in this thesis includes a model for learning which of the contents features in a tweet contribute most towards the *interestingness* of a tweet and empirically showed that this notion of *interestingness* can be used as a measure of static content quality in Twitter for retrieving high quality information (c.f Chapter 4). The second contribution is the FREuD approach for sentiment based social content diversification and we have experimentally showed that the FREuD approach (c.f Chapter 5) helps to find and recommend a subset of the product reviews that covers as many as possible aspects of a product and associated range of diversified user sentiments. Our third contribution in this thesis is the Author-Topic-Time model (c.f Chapter 6) for capturing temporal dynamics of latent topics and user interests in the social media and associated evaluation results from running the model on scientific publications dataset obtained from CiteSeerX.

The following sections lists the details of our findings in each of the contributions mentioned above.

## 7.1 *Interestingness*: a Measure of Static Content Quality

Chapter 4 answered two research questions that were established in Scenario 1.1. To answer the question “What is of interest on the Twitter?”, we have analyzed tweet contents and identified which of the low-level and high-level content features contribute towards the interestingness of a tweet. Low-level content features include tweet terms, emoticons, exclamation and question marks, presence of the URLs’, usernames, hashtags, etc., while high-level features include sentiment polarity and topic composition of the tweet. Based on the identified features, we trained a logistic regression model to predict which of the tweet features contribute to the likelihood of its retweetability. From this analysis we have drawn the following conclusions:

- Concerning the low-level features, a tweet is likely to be retweeted when it contains URLs, usernames, hashtags, negative emoticons and question marks, while it is less likely to be retweeted when it contains an exclamation mark or positive emoticons. A tweet is also unlikely to be retweeted when it is addressed to another Twitter user directly.
- A tweet is likely to be retweeted when it is about a general, public topic instead of a narrow, personal topic. This can be understood as the Twitter platform being better suited as a news and announcement channel.
- Sentiment polarity of a tweet also plays an important role for retweeting and we observe that a tweet is more likely to be retweeted when its sentiment polarity is negative, implying bad news travel fast.

To answer the second question of whether or not our notion of *interestingness* can be used as a measure of static content quality to retrieve high quality contents, we have looked at the problem of sparsity, the effects of document length normalization and content quality in Twitter. The feature sparsity is immanent to the restriction of the medium to short texts as each tweet can be of no more than 140 characters long. The features sparsity is problematic in retrieval as term frequencies are used for estimating the importance of a term in the document. In shorter texts it is nearly a binary value, thus, most retrieval models are effectively reduced to using global term weights, that measure the discriminativeness of terms. The quality assessment of the Twitter contents is necessary as the twitter documents range from spam over trivia and personal chatter to news broadcasts, self presentation, information dissemination, and reports of current hot topics. We have based our notion of *interestingness* on the retweet function of the Twitter and assumed retweet as an indicator of high quality content that

are of general interest on the Twitter. We used the retweet likelihood score to mark the interestingness of a tweet and used it in the retrieval process to retrieve and filter high quality tweets for a given user information need. The results from this analysis and experiments led us to following conclusions

- Our analysis across several large Twitter datasets have shown that about 85% of all Twitter messages contain each term at most once confirming that Twitter is inherently sparse.
- Document length normalization is counterproductive as it introduces an unmotivated bias towards short documents in microblog retrieval.
- *interestingness* when used as a measure of static content quality improved retrieval performance in the sense of providing more relevant and generally informative messages in the search results.

## 7.2 The FREuD Approach

In Chapter 5, we have looked at the problem of content diversification in social media as described in the Scenario 1.2. We motivated ourselves from the problem of finding a subset of product reviews that can best serve the user information need of obtaining a diversified view point about the product features. We identified three challenges in the diversification task, i.e. automatic product feature mining, estimating sentiment orientation and finding a strategy for selecting an optimal set of reviews that covers as many as possible features and associated diversified sentiments. For this purpose, we have developed the FREuD approach which provides a unified solution to the above mentioned challenges. We tested the FREuD approach with a real world dataset of product reviews collected from the CNET. With our analysis and experiments on the data using the FREuD approach, we have concluded that

- There is a linear to sub-linear relationship between review length and over all sentiment score of the review. Longer reviews tend to be positive, whereas negative reviews tend to be shorter in length.
- LDA topics provide a good approximation of the product features when obtained after pre-processing the reviews content, suggesting that it is possible to mine product features in an unsupervised way from review content using certain text pre-processing techniques.
- The performance of the FREuD variations that used sentiment word based length normalization of sentiment scores and no length normalization of sentiments scores were at par with each other and in general performed better than both baseline approaches, where one baseline

imitated the default review ranking mechanism which is based on the usefulness of the review as voted by other users in the CNET. While, The performance of the FREuD approach which used standard length normalization of sentiment scores in diversification task was worst of all the approaches, suggesting that there is no need for length normalization of sentiment scores to balance the effect of longer reviews.

### 7.3 Author-Topic-Time Model

In Chapter 6, we have analyzed the problem of modeling a topic life cycle and correlating it with user interests in the social media content and have provided a solution to the Scenario 1.3. To this end, we used a Bayesian approach of unsupervised learning and proposed a novel Author-Topic-Time model that extended LDA to incorporate the text, author and time information in the document generation process. From the evaluation results of our experiments on a research publications dataset from CiteSeerX, we are able to conclude that

- It is possible to jointly model contents, authors and time in a three dimensional model which is able to capture topic life cycle and user interests at the same time providing a better explanation of topic life cycles.
- In general, the Beta distribution is helpful in modeling the topic life cycle, however, it may not be suitable when a topic exhibit spikes in its life cycle.
- Time is the most sparse attribute of the document and modeling it as a continuous variable jointly with text and author information of the document increases the complexity of the model making it difficult to scale for large document collections.

### 7.4 Outlook

In our analysis of social media contents in three example scenarios, we propose the following lines of work that can be used to extend our approaches.

In the analysis of Twitter contents, we have ignored the social context of the user, the global network structure, contents of the web pages linked through URL and the time of the tweet. Enriching the analysis with these features may lead to an improved retweet prediction and thus leading to a more sophisticated measure of content quality. Our analysis of content quality could also be interesting in the field of measuring influence among Twitter users. Another interesting topic is that of spam. As spammers also use the retweet function to feign relevance of their messages, our methods

may be susceptible to spam. So far we employed only basic methods to filter out spam and more sophisticated methods might improve performance.

As far as our analysis of social content diversification is concerned, the further line of thought will be to refine the estimation of sentiments expressed about a given feature. Our current approach operates with a document global, coarse grained sentiment value which is broken down to the feature level. Using a more fine-grained detection of sentiments in document segments might allow for a more detailed annotation of features with sentiments.

With respect to our ATT model, the current approach does not scale well to large document collections. It will be interesting to see if the implementation of the Gibbs sampler specific to the ATT generation process solves the scalability issue or not. A topic during its evolution may exhibit spikes and the Beta distribution used to model topic life cycle is not able to capture such spikes if present. It will be a good idea to explore for distributions which can also model topic spikes to have a more precise view of the topic life cycle.



## Appendix A

# Data Set – the FREuD Approach

To evaluate our FREuD approach for social content diversification, we needed a test reference collection of product reviews annotated for product features discussed and the reviewers’ sentiments toward the features in the reviews. The purpose of this appendix is to provide details of the data collected and annotation process employed to obtain the test reference collection.

To this end, we used CNETs’ developer APIs<sup>1</sup> to collect the metadata available for each product under various product categories. Table A.2 provides the details of the attributes that were fetched from the CNET review portal for each product. From the information provided in the metadata, we used a screen scrapper to crawl the actual contents of the reviews. The information about the number of reviews collected for each product under various categories is given in Table A.1.

The complete dataset of end user reviews annotated for product features and associated sentiments is published in the form of xml files on <http://west.uni-koblenz.de/Research/DataSets/FREuD><sup>2</sup>. The dataset page also contains script files needed to compute various statistics about the data.

Table A.1: End user product reviews dataset annotated by the assessors for features and associated sentiments.

Category	# Products	# Reviews	# Features
Cell Phone	7	175	13
Camera	6	150	11
Printer	7	175	12

<sup>1</sup><http://developer.cnet.com/>

<sup>2</sup>The gold standard dataset obtained after the annotation process is available for further research use at <http://west.uni-koblenz.de/Research/DataSets/FREuD>.

Table A.2: Attributes available in metadata collected for each product using CNET developers’ APIs. This information is also published in the form of xml files in the gold standard dataset for each product.

<b>Product Level Attributes</b>	
<b>Attribute</b>	<b>Attribute Description</b>
<i>CategoryID</i>	Products on CNET are divided into various categories, for example cellphones, digital cameras, printers etc. This attribute returns the unique numeric id allotted to each category.
<i>ProductID</i>	Each product on CNET is given a unique number identifier. This field provides the unique identifier for each product.
<i>ProductName</i>	This attribute provides the complete name of the product.
<i>UserVotes</i>	This attributes records the total number of end user reviews available for each product.
<i>EditorRating</i>	In addition to end user reviews, CNET also provides an Editorial review of the product. This field contains the Editors’ rating of the product ranging from 1 to 10.
<i>UserRating</i>	Average end user rating of the product.
<i>ReviewURL</i>	URL of the product page listing the editor and end user reviews.
<i>Good</i>	Editor’s pick of positive features for the products.
<i>Bad</i>	Editor’s pick of negative features for the products.
<i>BottomLine</i>	Editorial summary of the product review.
<b>Review Level Attributes</b>	
<b>Attribute</b>	<b>Attribute Description</b>
<i>MessageID</i>	Unique identifier for each end user review.
<i>Stars</i>	End users’s rating of the product.
<i>Thumbsup</i>	Number of other users who find the given review as <i>Helpful</i> .
<i>Thumbsdown</i>	Number of other users who find the given review as <i>Not Helpful</i> .



```

assessor selects a product;
for each unassessed remaining review do
  randomly pick one review at a time and present it to the assessor
  side by side with the product specific preselected features;
  assessor reads review and ;
  for each feature (from the list): assessor checks do
    if feature is discussed in the review at all then
      for all found utterances discussing this feature do
        assessor ticks the appropriate option (positive, neutral,
        negative, both) to annotate the sentiment and polarity
        of the feature;
        mark the location of the utterance in the review;
      end
    else
      go to next feature;
    end
  end
end

```

**Algorithm 3:** Process used to obtain assessors feedback while developing gold standard.

To annotate the reviews, we used crowd sourcing approach and set up an online website for the assessors to participate in the annotation process. The Algorithm 3 describes the process used for presenting the reviews to the assessors and obtaining feedback. The screen shots provided below detail the information collected from the assessors during annotation process and include the assessors' identity (c.f. Figure A.1), the assessors' knowledge for the selected product (c.f. Figure A.2), the instructions provided to complete the annotation (c.f. Figure A.3), example task (c.f. Figure A.4) and the presentation format in which the actual reviews were shown to obtain the feedback on features and sentiments (c.f. Figure A.5).



Figure A.1: Screen shot of the FREuDs' evaluation website home page requiring assessor to input a unique id which was used to annotate each review by three unique assessors.

**F R E u D**

**WeST**  
People and Knowledge Networks

Home Contact

Thank you very much for joining us in help improving FREuD.  
Your feedback is very valuable to us.

**1. Gender:**

Male  
 Female  
 Don't ask

**2. Select product to read reviews?**

Products:

**3. Do you have prior knowledge/experience of the selected product?**

Yes  
 No  
 Little bit

**4. Do you own or have used the selected product?**

Yes  
 No

Copyright 2012 west.uni-koblenz.de. All Rights Reserved.

Figure A.2: Screen shot of the FREuDs' evaluation website assessor info page. The assessors were required to select a product from various product categories to annotate the product related reviews. Additionally assessors existing knowledge of the the selected product was also recorded.

**F R E u D**

**WeST**  
People and Knowledge Networks

[Home](#)   [Contact](#)

### Task Description and Example

**Task Description:**  
Dear Participant we will show you one user review for the selected product and your task is to:

- **Read** this **review**,
- **Find** which of the **features** (listed on the right side of the review box) are discussed in the review,
- **Identify** the **sentiment** expressed by the user towards the feature in the review,
- **Record** your **observation** in the table provided on the right side of the page.

**Note:**

1. When you have identified the sentiment for a given feature in the review then please copy the feature name and sentiment word for that feature and paste it in the text field provided for that feature under "**Motivation for Sentiment**" column.
2. If you find that reviewer's sentiment is both positive and negative for different aspects of a given feature, for example if the reviewer writes that screen size is good but screen resolution is bad then you can select the option **Both(+ & -)** under "**Sentiment**" column. Please also copy all those feature aspect names and sentiment words from the review text and paste in the text field under "**Motivation for Sentiment**" column. Please use : to separate multiple sentences in Motivation text field
3. If you require explanation of a feature (aspects of a feature) then just take the mouse cursor to the small question mark displayed next to the feature in the feature column.

Figure A.3: Screen shot of the FREuDs' evaluation website instruction page. This pages provides the details of annotation process and step by step guide for annotating the features and feature related sentiment.

Product:


Digital Cameras -- Sony Cyber-shot DSC-HX9V (Black)

Review	Feature	Is feature discussed in review?	Sentiment about the feature?	Motivation for sentiment?
<p>I got the HX9v as soon as it came out this spring. It's pretty much the perfect, top-notch pocket point-and-shoot. In fact, I carry it with me in my right-hand pocket at literally all times. I have taken several thousand photos with it so far, and several hours of video. I am very happy with my purchase. The video is amazing, with many options, zooming while shooting, stereo sound, great auto focus, fantastic in low light. Keep in mind that for the highest quality setting, 1080p/60 fps AVCHD, if you want to view it on a Mac, you'll need to have the latest version of iMovie '11, running on at least the latest version of Snow Leopard (OS 10.6.8). Of course, you can now upload these .MTS files directly to the various on-line sites such as Facebook, YouTube, and Google+/Picasa, and you can view them on your HDTV through the mini-HDMI out (which is great). But for saving and viewing on the Mac, the AVCHD modes introduce a couple of extra steps and are not simple drag-drop-and-play. If you don't have a Mac running the above specs, there are still some good workarounds out there. After a lot of searching I found "ClipWrap" to be the best option for dealing with these files. But overall, I can't say enough about just how fantastic the video on this camera is. I've been hundreds of feet away at a concert in the dark, and taken footage that makes it look like I'm sitting right in front of the performer in a well-lit space. With nice-quality stereo sound. And all of this often on "only" the best 1080 MOV setting, not even dialing it up to the top-notch AVCHD. It's amazing, really. The 16x zoom is fast and very good. Things get a little noisy when you crop and zoom in on a shot that you took at maximum zoom (basically blowing up a shot to the center 1/9th of the frame after shooting at max zoom). But that's to be expected. 16x is fantastic and I use this feature every single day for nature shots - particularly birds. There's also a very nice feature one turn of the dial from auto mode that mimics the shallow depth-of-field in DSLRs. It takes two shots, blurs one, and then stitches the two together so that the subject is crystal-clear but the background is nicely blurred. I have gotten some very nice shots of flowers and animals close up with this, though the subject does have to be pretty still for it to work right. Speaking of flowers, the macro capabilities of the auto mode on this camera are FANTASTIC. Reason in itself to buy the camera. I can be on a walk, with my very big dog pulling in one hand. I'll see a very pretty flower and tell him to "wait" and "sit", while using my other hand to slide the camera out of my pocket and turn it on. I just hold the camera up close, click, slide it back into my pocket, and we're off again. Whole process takes maybe 2 seconds (unless I choose to take a couple of angles, etc.). This has been the case for nearly every one of my "local flowers", "Yosemite wildflowers", and "super macro" shots in my Flickr account taken with this camera (basically everything shot in the past few months). My Flickr handle is RobertCross1 if you'd like to check out any of my shots to see for yourself. I generally don't use the in-camera panorama features. While they are certainly very good for what they are, you can get much better panoramic shots by shooting consecutive, overlapping stills and stitching them together in a software application like Photoshop Elements. I also don't use the in-camera HDR feature. It just isn't very well implemented and produces pretty strange looking shots, even for HDR. If you want HDR photos, you can exposure bracket with three shots in very quick succession</p>	Battery ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input checked="" type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	In fact, they're
	Body ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	
	Ease of Use ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	using my other hand to
	Exposure ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	you can exposure
	Flash ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input checked="" type="radio"/> Both (+ & -) <input type="radio"/> Neutral	I never, ever use the flash
	Focus ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	great auto focus,
	Photo Quality ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	very nice shots
	Shooting speed ?	<input type="radio"/> Yes <input checked="" type="radio"/> No	<input checked="" type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	
	Video Quality ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	video is amazing
	View Finder ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	Superior display screen
Zoom ?	<input checked="" type="radio"/> Yes <input type="radio"/> No	<input checked="" type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	fast and very good	

I got it, lets

Figure A.4: Screen shot of the FREuDs' evaluation website example task. The example task page provides an example of a completed task according to the instructions given on the instruction page.

# F R E U D



Home    Contact

**Product:**  
Tablets -- Asus Eee Pad Transformer (16GB, Wi-Fi)

Review

There are enough reviews with excellent specs so this is just my impressions of the Asus Transformer. I wanted something small and light weight and needed to occasionally type which points to a netbook, but I also wanted a tablet for entertainment reasons. These things are cool. The various bluetooth keyboards are decent, but seemed disjointed, needing something to sit the tablet on and essentially you had a tablet with a wireless keyboard that performed like a tablet and wireless keyboard. You have to interact with the touch screen for a lot of functions. When the Asus Transformer and Keyboard are joined the tablet behaves pretty much like a netbook making it a lot more functional than a separate bluetooth Keyboard, plus the Keyboard will recharge the tablet if needed. The optional USB and SD slots are great to use with a thumb drive or easily take the SD card from a camera and view on the tablet. I have to see if the printer will work.

The Transformer is not near as powerful as my laptop, but smaller and lighter and I love a tablet. I feel it is a lot better than a stand alone tablet if you need to occasionally do some heavy typing, need extended battery life or the optional SD and USB ports or if I do not need to bring the laptop on trips. Some other thoughts. I would not buy any tablet without some sort of memory card slot for storage or a micro HDMI port if there are other ones in the same price range. Various manufactures have accessories that can be purchased to enable these functions, but this is an added cost and few more things to carry around and lose. The "cloud" storage offered with many devices (including Asus) is nice, but would be very limiting without WIFI not to mention they seem to have a yearly cost associated cost with them after a trial period. If you have pictures or videos that you would like for a group of people to watch, the HDMI port is wonderful to hook into a flat screen TV not to mention to be used as a video player on vacation. A standard movie converted for a tablet or smartphone is about 500 - 800MB.

They look pretty good on a TV, not great, but beats lugging around a book of DVD's. I didn't really think that the camera would be of much use, but my daughter used the tablet and in 20 minutes made a short "power point" of the pets. It is nice to take a picture and attach to an email. No flash though. If you do not need the Keyboard with its various benefits, then any good tablet would fit the bill. The 7" models are great for portability, with Dad loving his 7" tablet. All in all I am very happy with the Transformer and Keyboard. .

**Summary:**

-Optional "excellent" Keyboard dock-Micro SD slot in tablet-Standard SD slot on keyboard-Tablet works well-Micro HDMI slot on tablet-Two regular USB ports on Keyboard dock-Awesome battery life when connected to Keyboard-Nice viewing angles

-Power cord may be a little short and a new one costs \$30 or more-Keyboard does not come with a power cord so you must use the one that comes with the tablet or buy a new one

Feature	Is feature discussed in review?	Sentiment about the feature?	Motivation for sentiment?
Availability of Applications	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Battery ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Camera ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Design ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Display Readability ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Ease of Use ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Network Connectivity ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Performance ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Portability ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Screen Resolution ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>
Storage ②	<input type="radio"/> Yes <input type="radio"/> No	<input type="radio"/> Positive <input type="radio"/> Negative <input type="radio"/> Both (+ & -) <input type="radio"/> Neutral	<input style="width: 100%; height: 20px;" type="text"/>

Save and Next Review    Example Page    Exit but Save Current

Figure A.5: Screen shot of the FREuDs' evaluation website actual task page. This page shows the review text from the assessors' selected product and table for recording observations about product feature and sentiment.

# Bibliography

- [1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM '09*, pages 5–14, New York, NY, USA, 2009. ACM.
- [2] O. Alonso and R. A. Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *Proc. European Conf. on Information Retrieval*, pages 153–164. Springer Berlin Heidelberg, 2011.
- [3] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.
- [4] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [5] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Machine Learning Research*, 3:993–1022, March 2003.
- [7] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on Twitter. In *Hawaii Int. Conf. on System Sciences*, pages 1–10. IEEE Computer Society, 2010.
- [8] M. M. Bradley and P. J. Lang. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical report, The Center for Research in Psychophysiology, University of Florida, 1999.
- [9] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

- [10] V. Bush. As We May Think. *The Atlantic Monthly*, 176:101–108, July 1945.
- [11] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.
- [12] B. Carterette. An analysis of np-completeness in novelty and diversity ranking. *Information Retrieval*, 14:89–106, 2011.
- [13] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in Twitter: the million follower fallacy. In *Proc. Int. Conf. on Weblogs and Social Media*, pages 10–17, New York, NY, USA, 2010. ACM.
- [14] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, pages 288–296. Curran Associates, Inc., 2009.
- [15] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, pages 288–296. Curran Associates, Inc., 2009.
- [16] A. Che Alhadi, T. Gottron, J. Kunegis, and N. Naveed. Livetweet: Microblog retrieval based on interestingness and an adaptation of the vector space model. In *Proc. Text REtrieval Conference (TREC)*, pages 1–12. National Institute of Standards and Technology (NIST), 2011.
- [17] A. Che Alhadi, T. Gottron, J. Kunegis, and N. Naveed. Livetweet: Monitoring and predicting interesting microblog posts. In *Proc. European Conf. on Information Retrieval Demonstrations*, pages 569–570. Springer Berlin Heidelberg, 2012.
- [18] A. Che Alhadi, S. Staab, and T. Gottron. Exploring user purpose writing single tweets. In *Proc. Web Science Conf.*, 2011.
- [19] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 429–436, New York, NY, USA, 2006. ACM.



- [20] V. Cheng and C. H. Li. Linked topic and interest model for web forums. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, pages 279–284, Washington, DC, USA, 2008. IEEE Computer Society.
- [21] M. D. Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, and A. Kelliher. How does the data sampling strategy impact the discovery of information diffusion in social media? In *Proc. Conf. on Weblogs and Social Media*, pages 34–41. The AAAI Press, 2010.
- [22] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 659–666, New York, NY, USA, 2008. ACM.
- [23] C. Cleverdon. Readings in information retrieval. chapter The Cranfield Tests on Index Language Devices, pages 47–59. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
- [24] C. W. Cleverdon. Aslib cranfield research project: report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. *Cranfield Library*, 1962.
- [25] C. W. Cleverdon. The Cranfield tests on index language devices. *Aslib Proceedings*, 19(6):173–194, 1967.
- [26] W. S. Cooper, F. C. Gey, and D. P. Dabney. Probabilistic retrieval based on staged logistic regression. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, pages 198–210, New York, NY, USA, 1992. ACM.
- [27] W. B. Croft and D. J. Harper. *Document retrieval systems*. Taylor Graham Publishing, London, UK, UK, 1988.
- [28] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 519–528, New York, NY, USA, 2003. ACM.
- [29] T. M. Department, T. Minka, and J. Lafferty. Expectation-propagation for the generative aspect model. In *In Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359. Morgan Kaufmann, 2002.

- [30] L. Dugan. Twitter to surpass 500 million registered users. [http://mediabistro.com/alltwitter/500-million-registered-users\\_b18842](http://mediabistro.com/alltwitter/500-million-registered-users_b18842), [Online; accessed 14-April-2013], Feb. 2012.
- [31] M. Efron. Information search and retrieval in microblogs. *Journal of the American Society for Information Science and Technology (JASIST)*, 62(6):996–1008, 2011.
- [32] M. Eirinaki, S. Pisal, and J. Singh. Feature-based opinion mining and ranking. *Journal of Computer and System Sciences*, pages 1175–1184, 2011.
- [33] J. Fleiss. *Statistical Methods for Rates and Proportions. Second Edition*. Wiley, John and Sons, Incorporated, New York, N.Y., 1981.
- [34] J. Fleiss et al. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.
- [35] K. Ganesan, C. Zhai, and E. Viegas. Micropinion generation: an unsupervised approach to generating ultra-concise summaries of opinions. In *Proceedings of the 21st international conference on World Wide Web, WWW '12*, pages 869–878, New York, NY, USA, 2012. ACM.
- [36] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741, 1984.
- [37] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.
- [38] T. Gottron and N. Lipka. A comparison of language identification approaches on short, query-style texts. In *ECIR '10: Proceedings of the 32nd European Conference on Information Retrieval*, pages 611–614, Berlin, Heidelberg, 2010. Springer-Verlag.
- [39] T. Griffiths. Gibbs sampling in the generative model of Latent Dirichlet Allocation. Technical report, Stanford University, 2002.
- [40] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.
- [41] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl 1:5228–5235, 2004.

- [42] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 78–87, New York, NY, USA, 2005. ACM.
- [43] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 491–501, New York, NY, USA, 2004. ACM.
- [44] H. Guo, H. Zhu, Z. Guo, X. Zhang, and Z. Su. Product feature categorization with multilevel latent semantic association. In *Proc. of CIKM*, pages 1087–1096, New York, NY, USA, 2009. ACM.
- [45] D. K. Harman. Overview of the first text retrieval conference. In *Proc. Text Retrieval Conference (TREC)*, pages 1–20, Washington, Nov. 1992. National Institute of Standards Special Publication.
- [46] G. Heinrich. Parameter estimation for text analysis. *Web: <http://www.arbylon.net/publications/text-est.pdf>*, 2:43, 2005.
- [47] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '99*, pages 50–57, New York, NY, USA, 1999. ACM.
- [48] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.
- [49] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in Twitter. In *Proc. Int. World Wide Web Conf.*, pages 57–58, 2011.
- [50] D. W. Hosmer and S. Lemeshow. *Applied logistic regression*. Wiley-Interscience Publication, 2000.
- [51] J. Huang, K. M. Thornton, and E. N. Efthimiadis. Conversational tagging in twitter. In *Proc. Conf. on Hypertext and Hypermedia*, pages 173–178, New York, NY, USA, 2010. ACM.
- [52] P. Ingwersen. *Information Retrieval Interaction*. Taylor Graham, London, 1992. <http://citeseer.ist.psu.edu/ingwersen92information.html>.
- [53] L. R. Janna Quitney Anderson. What is the potential future influence of big data by 2020? [http://elon.edu/docs/e-web/predictions/expertsurveys/2012survey/PIP\\_Future\\_of\\_Internet\\_2012\\_Big\\_Data\\_7\\_20\\_12.pdf](http://elon.edu/docs/e-web/predictions/expertsurveys/2012survey/PIP_Future_of_Internet_2012_Big_Data_7_20_12.pdf), July 2012.

- [54] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2):207–227, 2000.
- [55] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct. 2002.
- [56] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *WebKDD/SNA-KDD '07: Proc. of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, New York, NY, USA, 2007. ACM.
- [57] F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
- [58] M. I. Jordan. Graphical models. *Statistical Science*, 19:140–155, 2004.
- [59] M. Karam, T. Manos, d. R. Marten, and W. Wouter. Incorporating query expansion and quality indicators in searching microblog posts. In *Proc. European Conf. on Information Retrieval*, pages 362–367, Berlin, Heidelberg, 2011. Springer-Verlag.
- [60] E. Kim, S. Gilbert, M. J. Edwards, and E. Graeff. Detecting sadness in 140 characters: Sentiment analysis and mourning Michael Jackson on Twitter. Technical report, Web Ecology Project, Aug 2009.
- [61] N. Kobayashi, R. Iida, K. Inui, and Y. Matsumoto. Opinion mining on the web by extracting subject-aspect-evaluation relations. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 86–91. AAAI, 2006.
- [62] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [63] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [64] S. Kotz, N. Balakrishnan, and N. L. Johnson. *Continuous multivariate distributions. Volume 1. , Models and applications*. Wiley series in probability and statistics. J. Wiley & sons, New York, Chichester, Weinheim, 2000.
- [65] R. Krestel and N. Dokoochaki. Diversifying product review rankings: Getting the full picture. In *Proc. International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 138–145. Ieee, Aug 2011.

- [66] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.
- [67] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22:79–86, 1951.
- [68] D. Kuroepka. *Modelle zur Repräsentation natürlichsprachlicher Dokumente - Information-Filtering und -Retrieval mit relationalen Datenbanken*. Logos Verlag, Berlin, Germany, 2004.
- [69] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proc. Int. World Wide Web Conf.*, pages 591–600, New York, NY, USA, 2010. ACM.
- [70] P. S. Laplace. Memoir on the probability of the causes of events. *Statistical Science*, 1:364–378, 1986.
- [71] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.
- [72] C. Lin, Y. He, R. Everson, and S. M. Rieger. Weakly supervised joint sentiment-topic detection from text. *IEEE Trans. Knowl. Data Eng.*, 24(6):1134–1145, 2012.
- [73] J. Lin. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.*, 37(1):145–151, Sept. 2006.
- [74] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA, 2005. ACM.
- [75] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.*, 1(4):309–317, Oct. 1957.
- [76] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 171–180, New York, NY, USA, 2007. ACM.
- [77] N. Metropolis and S. M. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.
- [78] D. Mimno and D. Blei. Bayesian checking for topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 227–237, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

- [79] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, UAI'02, pages 352–359, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.
- [80] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.
- [81] R. Nagmoti, A. Teredesai, and M. De Cock. Ranking approaches for microblog search. In *Proc. Int. Conf. on Web Intelligence and Intelligent Agent Technology*, pages 153–157, Washington, DC, USA, 2010. IEEE Computer Society.
- [82] N. Naveed, T. Gottron, J. Kunegis, and A. Che Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proc. Web Science Conf.*, pages 1–8, 2011.
- [83] N. Naveed, T. Gottron, J. Kunegis, and A. Che Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proc. Web Science Conf.*, pages 1–8, 2011.
- [84] N. Naveed, T. Gottron, J. Kunegis, and A. Che Alhadi. Searching microblogs: Coping with sparsity and document quality. In *Proc. Int. Conf. on Information and Knowledge Management*, pages 183–188, New York, NY, USA, 2011. ACM.
- [85] N. Naveed, T. Gottron, S. Sizov, and S. Staab. Freud: Feature-centric sentiment diversification of online discussions. In *WebSci'12: Proceedings of the 4th International Conference on Web Science*, pages 321–326, 2012.
- [86] N. Naveed, T. Gottron, and S. Staab. Feature sentiment diversification of user generated reviews: The freud approach. In *Proc. of ICWSM-13: 7th International AAAI Conference on Weblogs and Social Media*, pages 429–438, Cambridge, MA, USA, 2013. The AAAI Press.
- [87] N. Naveed, S. Sizov, and S. Staab. Att: Analyzing temporal dynamics of topics and authors in social media. In *Proc. Web Science Conf.*, pages 1–7, 2011.
- [88] N. Naveed, S. Sizov, and S. Staab. Attention: Understanding authors and topics in context of temporal evolution. In *Proc. European Conf. on Information Retrieval*, pages 733–737, 2011.

- [89] F. A. Nielsen. A new anew: evaluation of a word list for sentiment analysis in microblogs. In *Proc. of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, pages 93–98, 2011.
- [90] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134, 2000.
- [91] J. R. Norris. *Markov chains*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 1998.
- [92] B. Pang and L. Lee. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
- [93] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
- [94] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [95] A. Pepe, H. Mao, and J. Bollen. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *CoRR*, abs/0911.1583, 2009.
- [96] S. Petrović, M. Osborne, and V. Lavrenko. The Edinburgh Twitter corpus. In *Proc. Workshop on Computational Linguistics in a World of Social Media*, pages 25–26, 2010.
- [97] S. Petrović, M. Osborne, and V. Lavrenko. Rt to win!predicting message propagation in twitter. In *Fifth International AAAI Conf. on Weblogs and Social Media*, pages 586–589. The AAAI Press, 2011.
- [98] M. Plummer. *Jags: A program for analysis of bayesian graphical models using gibbs sampling*, 2003.
- [99] M. F. Porter. An algorithm for suffix stripping. *Readings in Information Retrieval*, 14(3):130–137, 1980.
- [100] M. F. Porter. An algorithm for suffix stripping. *Readings in Information Retrieval*, 14(3):130–137, 1980.

- [101] L. Qiu, W. Zhang, C. Hu, and K. Zhao. Selc: a self-supervised model for sentiment classification. In *Proc. Conference on Information and Knowledge Management*, pages 929–936, New York, NY, USA, 2009. ACM.
- [102] P. Resnik and E. Hardisty. Gibbs sampling for the uninitiated. Technical report, University of Maryland, Oct. 2009.
- [103] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR Conf. on Research and development in information retrieval*, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
- [104] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and passivity in social media. *CoRR*, abs/1008.1253:113–114, 2010.
- [105] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.
- [106] T. Sakai and R. Song. Evaluating diversified search results using pertinent graded relevance. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1043–1052, New York, NY, USA, 2011. ACM.
- [107] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. Int. World Wide Web Conf.*, pages 851–860, New York, NY, USA, 2010. ACM.
- [108] M. J. Salganik and D. J. Watts. Web-based experiments for the study of collective social dynamics in cultural markets. *Topics in Cognitive Science*, 1:439–468, 2009.
- [109] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.
- [110] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [111] G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Commun. ACM*, 26(11):1022–1036, Nov. 1983.
- [112] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1983.



- [113] L. Shi and J. MingYu. A dfm model of mining product features from customer reviews. In *Control, Automation and Systems Engineering (CASE), Conf. Proc.*, 2011.
- [114] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proc. Int. Conf. on Research and Development in Information Retrieval*, pages 21–29, New York, NY, USA, 1996. ACM.
- [115] C. Smith. By the numbers 10 amazing twitter stats. <http://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/>, [Online; accessed 14-April-2013], April 2013.
- [116] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *Conf. on Computer Vision and Pattern Recognition Workshops*, pages 1–8, Los Alamitos, CA, USA, 2008. IEEE Computer Society.
- [117] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in Twitter network. In *Proc. Int. Conf. on Social Computing*, pages 177–184, Washington, DC, USA, 2010. IEEE Computer Society.
- [118] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and Web search. In *Proc. Int. Conf. on Web Search and Data Mining*, pages 35–44, New York, NY, USA, 2011. ACM.
- [119] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03*, pages 173–180, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [120] P. Tsaparas, A. Ntoulas, and E. Terzi. Selecting a comprehensive set of reviews. In *Proc. of the ACM international conference on Knowledge discovery and data mining*, New York, NY, USA, 2011. ACM.
- [121] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
- [122] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual In-*

- ternational Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM.
- [123] C. Wang, D. M. Blei, and D. Heckerman. Continuous time dynamic topic models. In *UAI'08*, pages 579–586, New York, NY, USA, 2008. ACM.
- [124] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 424–433, New York, NY, USA, 2006. ACM.
- [125] X. Wang, N. Mohanty, and A. McCallum. Group and topic discovery from relations and text. In *Proceedings of the 3rd international workshop on Link discovery*, LinkKDD '05, pages 28–35, New York, NY, USA, 2005. ACM.
- [126] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: Finding topic-sensitive influential twitterers. In *Proc. Int. Conf. on Web Search and Data Mining*, pages 261–270, New York, NY, USA, 2010. ACM.
- [127] Wikipedia. Vector space model. [http://en.wikipedia.org/wiki/Vector\\_space\\_model](http://en.wikipedia.org/wiki/Vector_space_model), [Online; accessed 14-April-2013], 2013.
- [128] Z. Zhai, B. Liu, H. Xu, and P. Jia. Clustering product features for opinion mining. In *Proc. International conference on Web search and data mining*, New York, NY, USA, 2011. ACM.
- [129] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Proc. European Conf. on Information Retrieval*, pages 338–349, Berlin, Heidelberg, 2011. Springer-Verlag.

# Curriculum Vitae

Nasir Naveed

## Contact Information

**Address** Institute for Web Science and Technologies  
University of Koblenz-Landau  
Universitaetsstr. 1  
56070 Koblenz - Germany

**Telefon** +49 261 287-2782

**e-Mail** naveed@uni-koblenz.de

## Research Interests

- Social Media Content Analysis.
- Web Mining.

## Education

from 2008	<b>Ph.D. Scholar in Computer Science.</b> Institute for Web Science and Technologies, University of Koblenz-Landau.
2004	<b>M.S. (M. Phil.) in Computer Science.</b> University of Agriculture, Faisalabad Pakistan.
2001	<b>M.Sc. in Computer Science.</b> University of Agriculture, Faisalabad Pakistan.
1998	<b>B.Sc. (Hons) Agri.</b> University of Agriculture, Faisalabad Pakistan.

## Research Projects

- **EU IP ROBUST**<sup>3</sup>, Risk and Opportunity management of huge-scale BUSiness communiTy cooperation, 2010-2013, WP5 - Community Analysis
- **EU STReP WeGov**<sup>4</sup>, Where eGovernment meets the eSociety, 2008-2012, WP2 - Analytics of Online Discussions

---

<sup>3</sup><http://robust-project.eu/>

<sup>4</sup><http://wegov-project.eu/>

## Academic and Professional Experience

- 03.2012 – 03.2013     **Researcher.**  
Institute for Web Science and Technologies,  
University of Koblenz-Landau.
- 05.2008 – 02.2012     **Research Assistant.**  
Institute for Web Science and Technologies,  
University of Koblenz-Landau.
- 08.2006 – 08.2007     **Assistant Professor (Computer Science).**  
Virtual University, Lahore Pakistan.
- 02.2002 – 08.2006     **Lecturer, (Computer Science).**  
University of Agriculture, Faisalabad Pakistan.
- 04.2001 – 02.2002     **Software Developer.**  
MMTech Pvt. Ltd., Lahore Pakistan.

## Publications

### Conference Publications

- N. Naveed, T. Gottron, and S. Staab. Feature sentiment diversification of user generated reviews: The FREuD approach. In: *Proceedings of 7th International AAI Conference on Weblogs and Social Media, ICWSM 2013*, pages 429–438, Cambridge, MA, USA, July 08-10, 2013.
- A. Che Alhadi, T. Gottron, J. Kunegis, and N. Naveed. Livetweet: Microblog retrieval based on interestingness and an adaptation of the vector space model. In: *Proceedings of Text REtrieval Conference, TREC 2011*, pages 1–12, Gaithersburg, Md. USA, Nov. 15-18, 2011.
- N. Naveed, T. Gottron, J. Kunegis, and A. Che Alhadi. Searching microblogs: Coping with sparsity and document quality. In: *Proceedings of International Conference on Information and Knowledge Management, CIKM 2011*, pages 183–188, Glasgow, Scotland, UK, Oct. 24-28, 2011.
- N. Naveed, T. Gottron, J. Kunegis, and A. Che Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In: *Proceedings of Web Science Conference*, pages 1–8, Koblenz, Germany, June 14-17, 2011.
- N. Naveed, S. Sizov, and S. Staab. ATT: Analyzing temporal dynamics of topics and authors in social media. In: *Proceedings of Web Science Conference*, pages 1–7, Koblenz, Germany, June 14-17, 2011.

## Poster und Demos

- N. Naveed, T. Gottron, S. Sizov, and S. Staab. FREuD: Feature-Centric Sentiment Diversification of Online Discussions. In: *Proceedings of Web Science Conference*, pages 321–326, Evanston, IL, USA, June 22-24, 2012.
- A. Che Alhadi, T. Gottron, J. Kunegis, and N. Naveed. Livetweet: Monitoring and predicting interesting microblog posts. In: *Proceedings of 34<sup>th</sup> European Conference on Information Retrieval, ECIR 2012*, pages 569-570, Barcelona, Spain, April 02-04, 2012. 2012.
- N. Naveed, S. Sizov, and S. Staab. Attention: Understanding authors and topics in context of temporal evolution. In: *Proceedings of 33<sup>rd</sup> European Conference on Information Retrieval, ECIR 2011*, pages 733-737, Dublin, Ireland, April 18-21, 2011.

## Programme Committee

- MAMA 2013: Workshop on Metrics, Analysis and Tools for Online Community Management, Sep. 16-20, 2013, Koblenz, Germany.
- ELDEC 2007: E-Learning and Distance Education Conference, April 16-17, 2007, Lahore, Pakistan.