Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

Yu Wang	Date

Modeling Temporal Dynamics in User Generated Content

By
Yu Wang

Doctor of Philosophy
Computer Science

Eugene Agichtein, Ph.D.
Advisor

Michele Benzi, Ph.D.
Committee Member

Jeffrey K. Staton, Ph.D.
Committee Member

Mark Dredze, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Modeling Temporal Dynamics in User Generated Content

By

Yu Wang
Bachelor of Science, Zhejiang University, 2009

Advisor : Eugene Agichtein, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science
2014

Abstract

Modeling Temporal Dynamics in User Generated Content

By Yu Wang

The evolving nature of user generated content (UGC) lays the key characteristics of Web 2.0. The evolution process in UGC offers valuable evidence to explain the content dynamics in the past and predict trends in the future. In this dissertation, we design models to analyze content evolution patterns of UGC in three granularities: words, topics and sentiment. More specifically, this dissertation investigates content evolution in the following aspects: (1) on word-level dyanmics: analyzing word frequency change in collaboratively generated content and using historical word frequencies to better weigh the words in ranking functions; (2) on topic-level dynamics: learning temporal transition patterns of topics in microblog streams and predict future topics according to historical posts; (3) on sentiment-level dynamics: estimating and understanding different sentiment change patterns of popular political topics across different user groups. We show that the developed models enable new applications in UGC, such as improving content-based ranking, anticipating future popular topics and visualizing and interpreting sentiment dynamics.

Modeling Temporal Dynamics in User Generated Content

By

Yu Wang
Bachelor of Science, Zhejiang University, 2009

Advisor : Eugene Agichtein, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science
2014

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The modern web allows and encourages the public to generate and share content. As every web user could become a content provider who creates or revises webpages, the web content has never been more dynamic. Popular websites supporting UGC have emerged and expanded in various domains, including encyclopedia (Wikipedia), microblogging (Twitter, Weibo), question answering (Yahoo Answers) and video sharing (Youtube). These platforms not only house massive amount of data for public to consume, but provides a portal for users to express opinions, spread news and share knowledge. Thus, the dynamics in UGC, such as word frequency change in versioned documents (Wikipedia), topic popularity and sentiment change in microblogs, contain valuable information about the growth of content, the evolution of public interests, and the drift of mass opinions. On the other hand, the dynamics of UGC bring challenges to traditional applications and opportunities to new ones. Models previously designed for static content either fail to describe dynamics or need to be improved by leveraging the pattern of changing in words, topics and sentiment. Learning from historical evolving patterns of content enables predictive models that anticipate trends and users's needs in the future. In this dissertation, we observe the patterns in temporal dynamics, design models to harness the evolving UGC, and use the learned knowledge to facilitate

applications.

1.1 Definition of User Generated Content (UGC)

Traditional static webpages are usually built and maintained by authorities and consumed by the general public. In contrast, UGC websites allow users to create and share content with other web users.

User generated content (UGC) is often considered to have the following characteristics: (1) created by the public; (2) meant for sharing; (3) publicly available over the internet. Representative types of UGC include collaboratively generated content, microblogging content, question-answering systems, image and video sharing platforms. Many of these sites are among the most popular websites worldwide, such as Wikipedia, Twitter, Youtube, etc. In this dissertation, we mainly focus on two types of UGC, namely collaboratively generated content (CGC) and microblogging content (MBC).

1.2 Dynamics in User Generated Content

User generated content constantly changes over time. Due to the diversity in UGC websites, content dynamics can be interpreted in different ways.

Collaboratively generated content (CGC): A CGC site allows users to create new pages and revise existing pages. A page on CGC websites is usually a collaborative effort by many users. For example, Wikipedia, the most popular CGC site, allows users to edit a page by adding relevant content or by rewriting existing sentences. Each time a user edits a page, a revision of the page is generated. The authoring history of a Wikipedia page consists of all its revisions generated over time. In this case, content dynamics can be interpreted as differences between the revisions of a page, and such

differences include word frequency change, restructured page layout, newly introduced hyperlinks or citations, etc.

As of 2012, Wikipedia has more than 4 million pages and a Wikipedia page has an average of 89 revisions. Popular pages are updated or revised more frequently and significantly, especially when relevant events are progressing. All the edits and revisions contribute to the dynamics in CGC.

Microblogging content (MBC): An MBC site lets users post short messages (the length of the message is often restricted) to their social networks. For instance, Twitter, one of the most popular microblogging services, lets users create and post Tweets (messages typically shorter than 140 characters) to the ones following them. Dynamics in MBC can be observed in two ways: (1) on user-level, dynamics occur when the content and topics in a personal Tweet stream change; (2) on topic-level, dynamics lie in the ongoing discussion (often participated by many users) of a particular issue or progressing events.

Since introduced in 2005, microblogs gain rapid growth and have become a major tool for opinion expressing and information sharing. As of 2013, Twitter has about 500 million users, and more than 400 million Tweets are generated per day, not even mentioning that Twitter is not the only popular microblogging service in the world. Active microblogging users post about their status on an hourly basis or even more frequently. Discussion of popular topics and events could draw participation from millions of users. All these users and their posts make microblogging platforms such an ever-changing environment.

We now show examples of content dynamics in both Wikipedia and Twitter occurred during the Supreme Court decision of "Defense of Marriage Act" (DOMA) case. The Wikipedia page of "DOMA" was created in 2002. On June 26th, 2013, U.S. Supreme Court struck down DOMA Section 3, and the page got revised 109 times on that day.

Below shows the difference in the first paragraph of the “DOMA” page before and after those edits.

> The Defense of Marriage Act (DOMA) is a United States federal law that restricts federal marriage benefits and required inter-state marriage recognition to only opposite-sex marriages in the United States... Section 3 of DOMA codifies the non-recognition of same-sex marriages for all federal purposes...

First paragraph of Wikipedia page “DOMA” **before** the Supreme Court decision.

> The Defense of Marriage Act (DOMA) is a United States federal law that allows states to refuse to recognize same-sex marriages granted under the laws of other states. Until **Section 3 of the Act was ruled unconstitutional in 2013**, DOMA, in conjunction with other statutes, had barred same-sex married couples from being recognized as "spouses" for purposes of federal laws, effectively barring them from receiving federal marriage benefits...

First paragraph of Wikipedia page “DOMA” **after** the Supreme Court decision.

After the Supreme Court decision of “DOMA” case, not only the sentences and structures in the corresponding Wikipedia page were significantly revised, but the word frequency has changed. The word “unconstitutional”, which is not present in the early revision, appears in the first paragraph of the page. The Supreme Court decision suddenly boosts the relatedness between the word “unconstitutional” and the “DOMA” page, which provides evidence for search engines to score “DOMA” page higher for the query “unconstitutional”.

Dynamics triggered by “DOMA” also appear in Twitter. Figure 1-1 shows the number of Tweets about “DOMA” and same-sex marriage over time. More than one million Tweets about “DOMA” are generated on the day of Supreme Court decision. We can

also see the bursts of “positive” and “negative” Tweets on that day, which suggests that many users express their opinions with sentiment leanings in the posts, and the dynamics in sentiment come with the evolving content. From such dynamics, we can estimate how the public react to the decision, which offers a valuable analytical tool for politics, policy makers and government agencies to understand mass opinions.

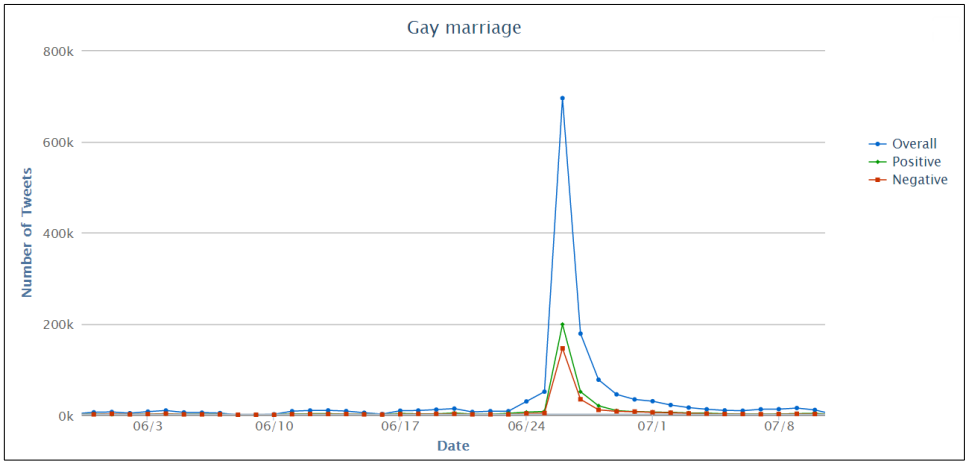

Figure 1-1: Volume of “Gay marriage” Tweets before and after the Supreme Court decision.

As UGC websites becomes extremely popular on the internet, the content has never been more dynamic. Users from all over the world can easily contribute to UGC by expressing their opinions, spreading news, and share their knowledge. However, such complex and evolving content can break into fundamental elements: words, topics, and sentiment. This dissertation aims to model dynamics of each elements and facilitate applications with the findings.

1.3 Motivation and Approach

Traditionally, content is often considered static in many existing models and algorithms, including ranking functions, generative topic models, and sentiment analysis. Such models often fail or are inadequate to describe UGC or utilize information contained in

UGC dynamics to facilitate other applications. This dissertation address the problem of modeling and utilizing content dynamics in the following areas:

1.3.1 Word Dynamics

As the most fundamental building block of text content, words are essential for many text-based applications, such as document ranking, clustering, classification, etc. Compared to traditional static documents, words in CGC have changing frequencies over time. Word frequency in a document has been heavily used to determine the relevance between words and documents in many static ranking functions. This dissertation proposes Revision History Analysis model to address the challenges brought by changing word frequencies in static ranking functions.

Problems Ranking CGC for search queries is currently treated in the same way as ranking static documents. This approach effectively ignores the actual document generation process, which could contain valuable information about relevance changes between words and documents. The process of content authoring is a collaborative effort of human editors, which reflects their knowledge about the world, as well as their judgment of the relative importance of words for a given topic. Using the generating process of the content to design ranking models would offer the opportunity to capture the meaningful change of the content and filter out the noisy ones.

Models We propose Revision History Analysis (RHA) to weigh a word in a versioned document by not only its frequency in the latest snapshot, but the frequency history of all revisions. RHA assumes that important words would appear early in the revision history and persist over time. Additionally, RHA also models the sudden change of word frequency to capture the time-sensitive relevance change between words and documents.

Implications With new word weighing model defined by RHA, we extend existing ranking functions, such as TF-IDF, BM25, language models, to be able to incorporate word dynamics into ranking CGC pages. Moreover, our analysis shows that frequent words in early revisions are more likely to have increased frequencies after revision. The results could help build a predictive model to anticipate word frequency change in the future, and therefore facilitate word-based applications for versioned documents.

1.3.2 Topic Dynamics

Traditional topic modeling techniques are mostly designed for static corpus. Recent dynamic topic models [14, 86] focused on inferring topic popularity change over time. This dissertation proposes temporal-LDA to describe the topic change from a new perspective: topics in personal Tweet streams transit from one to another, and the transition patterns can be learned from historical Tweet streams.

Problems Topics in UGC evolve as the content changes. Especially in personal Tweet streams, a user tends to Tweet about different issues rather than repeats the same topic over and over. Modeling topic dynamics provides an opportunity to learn how topics change and predict users' needs and interests in the future. By leveraging the Tweeting history of a user, we can learn how topics transition from one to another in the Tweet stream, and estimate the likelihood of possible future topics in one's Tweets. A successful prediction of users' future interests would better support content-aware and targeted recommendation and advertising.

Models We propose Temporal-LDA (TM-LDA) to learn topic transition pattern from adjacent Tweets in individual Tweet streams. First, we infer topics of each Tweet in the collection by traditional LDA. And then a transition matrix is computed according to the topic assignment of adjacent Tweet pairs.

Implications The learned transition patterns can help anticipate topics of future Tweets given users' most recent posts. The prediction is content-aware and the results can help build intelligent recommendation systems. Also, by investigating topics that precede or proceed targeted issues, the transition patterns can help us understand and identify potential causality of social phenomenons.

1.3.3 Sentiment Dynamics

Sentiment dynamics can reflect public opinions and reactions to popular issues or events. Existing sentiment analysis techniques in microblogging content focus on individual posts or users, which often produce estimated polarity for a single user or mixed sentiment trends of all users. The challenges rise when we try to identify dynamics generated by a group of users with certain characteristics (e.g., "African-American middle-age female"). We propose an unsupervised learning technique to infer latent user characteristics from their self-descriptions and social networks, and then estimate sentiment change of users in different characteristic groups.

Problems The sentiment of UGC changes with the content. As microblogging has become a major tool for people to express and amplify opinions, sentiment dynamics in microblogs offers a great opportunity to measure public reaction to major events, political issues, commercial products, etc. While raw sentiment and opinions are important, they are only part of the story. For information to be meaningful it must be contextualized within a socio-economic and cultural frame. To contextualize sentiment and its dynamics, we need not just what the opinion is, but who has it. Understanding users' characteristics and segmenting users into meaningful characteristic groups are essential to truly understand the sentiment dynamics in MBC.

Models Our goal is the unsupervised discovery of user characteristics based on available data for each user. We propose a generative model to infer user characteristics directly from their self-descriptions in social media profiles. The proposed model assumes words in self-descriptions are generated by latent characteristics of users. While user descriptions can be highly informative to reveal characteristics, they are not always so. We leverage users' social network information (i.e., who they follow) to more robustly infer characteristics even when the users lack descriptions.

Implications Understanding users by inferring latent user characteristics is valuable to most opinion mining applications. Since the user is a key element in MBC, knowing user characteristics can help divide the mixed and aggregated sentiment and opinions into coherent opinion groups, and understand concerns or emotions from different types of users. Moreover, user characteristics can help learn the pattern of sentiment dynamics generated by distinct user groups, and potentially facilitate predicting user reactions to future issues and events. Experiments show that our automatically inferred user characteristics outperform traditional demographic categories in estimating Twitter users' political sentiment, which suggests a new way to understand public opinion.

1.3.4 Temporal-Dependent User Behavior

Users are the key element in user generated content, especially in microblogging platforms. They post messages, make connections, and promote content generated by others. All these actions and behavior shape the microblogging sites as dynamic as they are today. The temporal orders of those actions and behavior could contain valuable information, such as users' interests and the importance of the links in their social networks. We adapt our previously introduced user characteristic inference model to investigate how user behavior depends on time on characteristic-level. The findings improve the quality of the inference of user characteristics and have implications for other applica-

tions.

Problems Users' actions at a specific timestamp or in a particular temporal order could imply valuable information about the user and his preference. For example, a user posts a related tweet when a breaking event happens could imply his interests to that event; The temporal order of building up the following network could reveal the perceived importance of the links. Yet, these underlying correlations between time and users' behavior (e.g., posting and following) have not been fully investigated in previous work.

Models We first study temporal-dependent posting behavior. We develop UserTime model to leverage both user's profile words and the time they post Tweets for user characteristic inference. Experiments show that the time of users' tweets could better infer users' unrevealed characteristics in self-descriptions and improve sentiment estimation. Second, we investigate the temporal following actions on cluster-level, where clusters are formed by automatically discovered characteristics. The results show that temporal following information gives better signals to estimate retweeting actions than overall following probabilities between clusters.

Implications The timestamps of users' posts helps recovering users' unrevealed characteristics. Our proposed UserTime model could potentially enable user profile completion and more accurately measure their sentiment. The temporal order of establishing following relationship in Twitter is better correlated with the number of retweets than overall following probabilities. Results suggest that early created links tend to carry more retweets and therefore offer more signals of users' information preference. This finding can help weigh the links with temporal information and impact network-based applications.

1.4 Applications

This dissertation not only describes temporal dynamics with presented models, but also take efforts to let users see and understand the dynamics through applications powered by proposed techniques. Particularly, we built a visualization and analytical tool to visualize topic and sentiment dynamics in MBC.

Dynamics in UGC are complex and involve many factors: words, topics, sentiment, users, etc. To let researchers and common users understand and investigate dynamics according to their needs, a visualization and analytical tool is highly demanded. We develop courtometer.com to display popularity and sentiment changes for pre-defined political topics. Users can also refine the topics by adding filters and constrains to our collection of data and produce highly customized "subtopics" in real-time. The website adopts and modifies models developed in this dissertation to improve run-time efficiency. Visualization results and the data can be downloaded and shared to facilitate further analysis and support research in other domains.

1.5 Dissertation Contributions

Overall, in this dissertation we model dynamics of words, topics, sentiment, and user behavior in UGC and use the findings to facilitate applications such as ranking versioned documents, predicting future topics, and more accurately classify sentiment. The main contributions of this dissertation include:

- We introduced Revision History Analysis (RHA) to weigh words according to the document edit history. RHA directly captures the document authoring process when available, and is particularly valuable for analyzing collaboratively generated content, notably Wikipedia documents. RHA can be naturally incorporated into state of the art retrieval models, as we demonstrate by showing consistent

improvements that RHA enables for BM25 and language models. (Chapter 3)

- We presented a novel temporal-aware language model, TM-LDA, for efficiently modeling streams of social text such as a Twitter stream of an author, by modeling the topics and topic transitions that naturally arise in such data. To capture the topic transitions in real-time, TM-LDA is designed in an online fashion and update the transition patterns as new Tweets emerge. We have shown that our method is able to more faithfully model the word distribution of a large collection of Twitter messages compared to previous state-of-the-art methods. (Chapter 4)

- We developed an unsupervised learning model to infer latent users characteristics from their self-descriptions and social networks on microblogging sites. The inferred user characteristics more accurately estimate user's sentiment towards popular political issues than human-annotated demographic attributes, which have been widely used to estimate public opinion in social science. (Chapter 5)

- We analyzed the relationship between time and users' behavior (posting and following) on microblogging websites. For posting behavior, we developed User-Time to make use of the time of users' posts to help inferring user characteristics. The resulting characteristics improved the performance of user profile completion and the accuracy of sentiment classification over the baseline model. For following behavior, we found that the temporal order of link creation helps better estimate users' information preference than overall following probabilities. (Chapter 6)

- We built *courtometer.com*, to visualize dynamics in MBC and help identify and analyze dynamics of customizable political topics. Techniques developed in this dissertation are adapted and incorporated into the website. (Chapter 7)

The dissertation introduces complementary approaches to addressing the common problem of modeling temporal dynamics of user generated content. Together, the proposed techniques already resulted in significant contributions in versioned document ranking, predicting future topics in microblogging streams, and analyzing sentiment trends of different user groups, and are likely to lead to new techniques and enable novel applications in interpreting and anticipating dynamics of user generated content.

Chapter 2

Background and related work

Dynamics in UGC include and trigger the changes in multiple dimensions, including relevancy, topic and sentiment dynamics. Words are the most fundamental elements to infer relevancy in most content-based retrieval models. The direct result of content dynamics would be the varying word frequencies in documents. Related work about ranking static and dynamic content is reviewed in Section 2.1. Latent topics inferred from the content would naturally change with the evolving content. Related work about modeling static and dynamic latent topics is reviewed in Section 2.2. Sentiment dynamics in UGC, especially in social media sites, come with the progressing events and evolving opinions. Modeling sentiment and sentiment dynamics is reviewed in Section 2.3. Understanding users' attributes has its irreplaceable role in interpreting content dynamics since all dynamics are generated by users. We review existing models of inferring users' (demographic) attributes in Section 2.4.

2.1 Ranking and Term Weighting for Search

Content-based ranking models have been intensively studied for decades. To retrieve and rank relevant documents to a given query, most ranking models infer the weights of

terms appearing in the query, and score documents based on the term frequencies. [59] has a comprehensive review of popular ranking techniques, including boolean models, vector-space models and probabilistic models. Representative models include BM25 [72] and statistical language models [51] etc. Several alternative directions for term weighting have been previously explored in the literature. Zaragoza et al. [73] proposed BM25F, a variant of BM25 [72], which computes term weights depending on which part (field) of the document the term appears in. Similarly, Ogilvie and Callan [63] used structural markup to combine document representations within the language modeling framework. Additional relevant approaches focusing on document structure include the work by Trotman [81] and Wang and Si [85]. Other approaches explored the use of statistical information of term occurrence in datasets other than the target retrieval corpus. Bendersky and Croft [10] identified key concepts in long queries using term frequency in Google's 1 Terabyte N-gram corpus, as well as in a large query log. Subsequently, Lease [53] also studied term weighting for verbose queries within the Markov random field model. Although these studies focused on query-side term weighting, in principle similar approaches could be applied to document-side term weighting as well. Other studies also modified the standard language modeling approach by considering relationships between words in a document [19]. A number of studies also developed new term weighting methods using topic models and cluster-based language models [50, 57, 87].

Yet, most traditional ranking models are designed for static documents, which basically assign term weights according to term frequencies in the static documents. The proliferation of UGC brings the research challenge of ranking evolving content. The definition of frequency-based term weighting becomes less clear in evolving content since term frequencies change over time. Several prior studies propose new ranking models for dynamic content by leveraging the history of the webpages. Adar et al [1] tried to distinguish between static and dynamic content on popular webpages. Elsas and Dumais [27] studied the dynamics of document content with applications to document

ranking. In their work, terms are categorized into three groups (long-term, mid-term and short-term) based on the length of time they are present in the documents, and then assign term weights accordingly. Efron [26] also used temporal information for determining term weights, yet he considered the change over time of the entire collection, rather than of individual documents. Thomas and Sheth [79] investigated the content dynamics in Wikipedia where they modeled each revision of an article as a term vector.

Our work propose a general term weighting model which incorporates authoring history of UGC and the changing pattern of term frequencies in dynamic content. The proposed model can be naturally incorporated into family of probabilistic retrieval models, and improve the ranking performance.

2.2 Topic Modeling

Topic models assumes the generation of documents is from mixtures of topics, and topics are represented by probability distribution over vocabularies. Intuitively, topics are more natural and general to human beings than words. Mapping content to topics enables indexing and searching documents in semantic dimensions. Since introduced in 1990's, topic modeling has emerged as one of the most effective methods for classifying, clustering and retrieving textual data. Representative algorithms include latent semantic analysis [25], probabilistic latent semantic analysis [43] and latent dirichlet allocation (LDA) [15] etc. Many models were then developed by extending LDA to better fit a particular corpus and incorporate more factors into the generation process of documents. For example, Purver et al. design a topic model to learn semantics in multi-part discourse and segment the meeting transcripts. Rosen-Zvi et al. [74]assume topics generated by an author are consistent across documents, and they extended LDA to include authorship information to infer topics. Griffiths et al. [37] argue that the generation process of words in an article depends on both latent semantic topics and local

syntactic structures.

To model topics in evolving content, dynamic topic modeling techniques are proposed. Comparing with static topic modeling algorithms, there are two major assumptions in dynamic topic models: First, there are temporal dependencies in topics and topic popularity; Second, weights between topics and words may change over time. Most dynamic topic models are extensions of LDA [14, 60, 86, 36, 84]. Blei and Lafferty [14] developed a Markov-chain style topic dependency model for temporally discretized corpus (e.g. yearly journals), and each snapshot of the corpus is modeled by LDA. Wang et al. [86] try to model the topics continuously over time. Their work treated topics as distributions over the actual timestamp of documents instead of discrete snapshots. It is worth to mention that many dynamic topic models are designed particularly for online or streaming text [5, 46, 34, 42, 75]. Since dynamics in UGC mostly come in as text streams (e.g., newly generated Tweets), these group of models can be adapted to model topic dynamics in UGC.

Topic modeling in UGC is not a trivial extension of modeling topics in traditional content [44]. First, text in UGC, especially in microblogs, can be very short and noisy. It's relatively difficult to observe enough signals (e.g., co-occurrence of words) and infer meaningful topics. Second, topics in UGC evolves so fast that it requires topic modeling algorithms to capture emerging topics as early as possible. Third, UGC contains rich meta information which could potentially help inferring topics. For example, it's easier to estimate the topics of an author's Tweets if we know his self-description, interests, location and who he is following. On the other hand, topic modeling in UGC enables and facilitates many applications, such as event tracking [55], trend detection [7] and popularity prediction [77]. Asur and Huberman [6] show that Tweets can help predict movie revenues. Paul and Dredze [66] suggest Tweets reflect public health information.

Our work views topic dynamics in UGC as topic transitions, where the topics in the future are transitioned from topics in the past by following explicit transition probabili-

ties.

2.3 Sentiment Analysis

General sentiment analysis has made significant advances over the last decade [65, 64, 56], and with the focus on certain aspects, such as intensity [89], polarity [88] and irony detection [20].

Sentiment analysis over UGC such as Twitter remains a challenging research topic [71, 20, 2, 13]. The noisy nature of Twitter content makes it difficult to classify short text fragments into sentiment classes. Barbosa and Feng [8] proposed to aggregate sentiment from several weak classifier and generate better sentiment classification results. Other work tried to use signals unique to microblogs, including social networks, hashtags, emoticons etc. Tan et al. [78] leveraged social network information to infer user-level sentiment. The assumption is that connected users may share similar opinions. Davidov et al. [23] argued that sentiment hashtags and emoticons are good indicators of sentiment leaning in Tweets.

UGC evolves over time, and so does the sentiment in UGC. However, the topic of sentiment dynamics has not been fully explored. Nguyen et al. [61] proposed to model and predict aggregated sentiment "direction" in Twitter. Guerra et al. [39] modeled public sentiment changes during popular events. They adjust the sentiment results by considering biased emotion disclosing behavior.

Our work relies on Twitter users' self-descriptions and their social networks to infer user characteristics, and then estimate sentiment with automatically discovered characteristics. Instead of reporting mixed and aggregated sentiment from all users, we can segment users into coherent opinion groups according to their characteristics, and observe sentiment and its change in each user group. By doing so, raw sentiment is contextualized with users' characteristics, which in turn help interpret and understand

the cause of sentiment dynamics.

2.4 Twitter User Classification and Attributes Inference

Understanding the characteristics of Twitter users can be challenging because people do not often report their age, gender, and other demographic attributes on Twitter. Many models have been proposed to infer missing user attributes, such as gender [16, 12, 90], age [90], religion [62], political leanings [58, 67, 21], etc. Rao et al. [70] used traditional n-gram features and statistics of Twitter social network features (e.g., follower and following count) to estimate gender, age, regional origin and political orientation. Zamal et al. [90] proposed to infer user attributes by leveraging actual content (Tweets) from neighbors in the network. Makazhanov and Rafiei [58] built a political orientation classifier for Twitter users, which mainly used the frequency of user engaging in political topic disscussions. Pennacchiotti and Popescu [68, 67] developed a model that leverages users' profile, tweets and social network information to classify partisan of Twitter users. Their model applied manually crafted regular expressions on users' self-descriptions to extract attributes such as gender and ethnicity. However, their method delivered very poor performance, and they claimed that the self-description fields do not contain high-quality information to be directly used for user classification purposes.

Unlike previous models, our work proposes a new way of using self-description content: inferring latent characteristics from users and their friends' self-descriptions with unsupervised topic models. Instead of inferring established demographic attributes, our method outputs a number of automatically identified user characteristics. Analysis shows that the resulting characteristics subsume traditional demographic categories and outperform demographic attributes in political sentiment analysis.

Chapter 3

Modeling Word Dynamics in Collaboratively Generated Content

Words are the essential building blocks of content for ranking, classification, and clustering documents. Word dynamics, i.e., word frequency change, would cause dynamics in relevance and ranking results. In this chapter, we propose the model, namely Revision History Analysis or RHA, to redefine term weighting based on term frequency in all revisions of the documents. We show that RHA provides consistent improvements for both BM25 and language model-based retrieval models on standard retrieval tasks and benchmarks.

3.1 Word Dynamics in Wikipedia

Wikipedia allows users to edit pages and generate revisions. A valid revision usually improves the previous one with more relevant context. Word dynamics naturally come with the generation of revisions. In this chapter, we consider the historical revisions of a document as a linear sequence of edits, ignoring special cases of such as reverting revisions to recover from vandalism. Thus, a page editor modifies a document by adding

relevant information or deleting non-relevant information from the previous version of this document, and generates a new version. This process suggests that relevant terms for a web document will frequently appear in most revisions and are rarely deleted. The frequency of these important terms is likely to grow along with the growth of the document length. On the other hand, non-relevant terms may exist in some revisions incidentally, but will be removed by editors in subsequent revisions. The main observation of RHA model is that the importance of a term in a document can be measured by analyzing word dynamics in the revision history.

3.2 Revision History Analysis

Our hypothesis is that the term weight for a versioned document should incorporate the term frequency in both the historical versions of the document, and in the latest (current) version of the document. For documents that grow incrementally (that is, following a steady expansion process), this model is sufficient, and is captured by our "global" model. However, some documents undergo series of dramatic changes, as the document is expanded or revised to reflect news events or significant bursts of editing effort, requiring our model to account for such significant changes in the document content. Thus, RHA model of term frequency incorporates the "global" term growth, the "bursty" document generation model, and the final (latest) version of the document at the time of indexing.

3.2.1 Global Revision History Analysis

We now introduce our first (and simplest) RHA model, which assumes that a document grows steadily over time.

Consider a document d from a versioned corpus D, and $V = \{v_1, v_2, ..., v_n\}$ to be the revision history of d. The number of revisions of document d is n. The latest revision of

document d is designated to be the latest document snapshot, $v_n \triangleq d$. Finally, let $c(t, d)$ be the frequency of term t in d.

We can now introduce a term weight according to the RHA global model, $TF_{global}(t, d)$, that would capture the appearance of t across the sequence of document versions. Intuitively, we wish to support the varying term importance across revisions, for example, to capture the importance of the few original terms used to describe a concept in Wikipedia, compared to terms added later in the document's "lifespan". Specifically, we define the new term weight as:

$$TF_{global}(t, d) = \sum_{j=1}^{n} \frac{c(t, v_j)}{j^{\alpha}}, \tag{3.1}$$

where j is the counter enumerating all revisions of document d from the first revision ($j = 1$), to the last revision ($j = n$). The raw frequency of term t in revision j is indicated by $c(t, v_j)$, is modified using the *decay factor* j^{α}, where α controls the speed of the decay. This decay factor, j^{α} adjusts the relative term weight across the multiple revisions to reflect the importance of term appears in different stages of the document evolution. For example, when $\alpha > 0$, the weight of a term will decrease in later revisions, to reflect the importance of a term appearing early in the document lifetime. In contrast, when $\alpha < 0$, the decay factor rewards the terms appearing in the latter revisions. In our experiments, we found that the optimal value for α was 1.1, implying that the term is more important if it appears early in the revision history of a document.

3.2.2 Revision History Burst Analysis

Documents can undergo intensive editing or massive content change when the popularity of a document increases, or when related events happen, which are immediately described on the document. We call such situations *bursts*, and extend the "global" decay model described above to capture these kinds of document evolution. These bursts are significant since the topic of a document may be different after the burst. For exam-

ple, the content of a document may be updated to reflect the latest news, and as a result the topic of the document can shift over time, as the news evolve.

For example, consider a Wikipedia page devoted to the movie “Avatar”. In the earlier (ca. June 2006) revisions of the page, there was little editing activity and little content, the page simply mentioned that James Cameron would direct the film, which was going to be released in 2009. However, in October 2006, there is a dramatic change to the content as new details about the plot, budget, and development are added. There is another “burst” in December describing the production and more details about filming. However, in the Wikipedia page that describes the meaning of the Hindu concept “Avatar”, its etymology and associated deities, the addition of content increment is considerably slower and editing bursts are much less frequent than in the movie-related page. Consequently, in the movie page, term weights are adjusted by incorporating burst history. In what follows, we present the RHA burst model, and we describe how to detect these bursts.

Recall, that the main assumption underlying the RHA model is that important terms are introduced early in the life of a document. However, a burst “resets” the decay clock for a term, in a way it “renews” the importance of the term. The intuition here is that if the term is still around after a major rewrite of the document content, then this term must be important. Note that a document could have multiple bursts over its revision history, as can be captured naturally in our approach.

Let $B = \{b_1, b_2, ..., b_m\}$ be the set of *burst indicators* for document d, and $m = |B|$ is the number of bursts. The value of b_j is the revision index of the end of the j-th burst of document d. We define the term weighting for the burst model to be the sum of decayed term weights over all the detected bursts. Each burst "boosts" the term weight for a short time, which then decays just like in the global model. Then, the overall term frequency weight of t is defined as:

$$TF_{burst}(t,d) = \sum_{j=1}^{m} \sum_{k=b_j}^{n} \frac{c(t,v_k)}{(k-b_j+1)^{\beta}}, \tag{3.2}$$

where k is the counter enumerating the revisions after burst b_j for each j. The raw frequency of term t in revision j, $c(t,v_k)$ is divided by the decay factor $(k-b_j+1)^{\beta}$. Thus, when a burst b_j happens, the decay factor for burst b_j will be set[1] to 1, and then the impact of this burst will gradually decrease in subsequent revisions because the decay factor increases with the growth of k. For a document d, Equation 3.2 calculates the impact of a burst by summing up term frequency with an exponential decay and adding the impacts of m bursts together. In our experiments, the optimal value for β estimated on the training set was found to be 1.1, equal to the value of α introduced in the preceding section.

For more convenient and effective manipulation we can also represent our burst model in a matrix form. Recall that the decay clock will be reset after each burst event - thus contributing a respective decay factor for each subsequent revision. These can be intuitively represented in a *decay matrix* $\mathbf{W}$, where each row i represents a potential burst position, and each column j represents a document revision. Each entry in W is computed as:

$$\mathbf{w}_{i,j} = \begin{cases} \dfrac{1}{(j-i+1)^{\beta}} & \text{if } i \leq j, \\ 0 & \text{otherwise.} \end{cases}$$

[1]Notice that $(k-b_j+1)^{\beta} = 1^{\beta}$ for the first revision after burst b_j.

Thus, the matrix W has the following structure:

$$\mathbf{W} = \begin{bmatrix} 1 & 1/2^{\beta} & 1/3^{\beta} & \dots & 1/n^{\beta} \\ 0 & 1 & 1/2^{\beta} & \dots & 1/(n-1)^{\beta} \\ 0 & 0 & 1 & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & & 1 \end{bmatrix} \tag{3.3}$$

where β in the the global parameter/exponent of the decay, and the i-th row of $\mathbf{W}$ corresponds to the set of the decay factors for due to the i-th burst in the editing history. If the only burst in the editing history occurred at revision v_i, the decay factors for the subsequent revisions, are stored in the cells $w_{i,i}, w_{i,i+1}, \dots, w_{i,n}$. The corresponding values are $1, 1/2^{\beta}, 1/3^{\beta}, \dots, 1/(n-i+1)^{\beta}$. Note that the matrix W is triangular, since bursts do not affect any revisions prior to a burst - that is, the columns to the left and above of the cell representing the burst event are not affected by the burst.

Of course, multiple rows in W could be associated with a burst, but probably not all rows – resulting in many *potential* burst positions. Thus, we introduce a vector $U = [u_1, u_2, ..., u_n]$ as the *burst indicator* vector that will be used to "filter" the decay matrix W to contain only the true bursts (we discuss how the bursts are detected in the next section). Specifically, each entry in U, u_j, is set to 1 if a burst is detected at revision j, and is set to 0 otherwise.

We now multiply the row vector U and the decay matrix W, results in a vector UW. Each entry of UW, $Uw_{*,j}$ contains the sum of the decay factors, each one set accordingly to the non-zero respective bursts prior to, and including, the j-th revision. For example, consider the case where $U = [1, 0, 1]$, that is, there was a burst detected in both revisions 1 and 3 but not in revision 2. Then, $uw_3 = 1 \cdot 1/3^{\beta} + 0 \cdot 1/2^{\beta} + 1 \cdot 1$, where the first term is the decay factor from the first burst, the second term is 0 since there was no burst in revision 2, and the third term is 1 since the burst is detected in the

current revision and this version's contribution is not yet decayed.

Thus, the term weighting of the RHA burst model for a document d can be finally computed as a scalar dot product between the vector UW and the term frequency vector C, where each entry $c(t, v)$ represents the raw frequency of term t in revision v of the document d. Specifically:

$$TF_{burst}(t, d) = UW \cdot \begin{bmatrix} c(t, v_1) \\ c(t, v_2) \\ c(t, v_3) \\ \vdots \\ c(t, v_n) \end{bmatrix} \tag{3.4}$$

where UW is the product of the burst indicator vector U with the decay matrix W as described above. The resulting modified term frequency value for a term t and document d, $TF_{burst}(t, d)$, combines the *decayed* values of term frequencies of t across all bursts in the edit history of d. For example, consider the case where d had only three revisions, the burst vector is $U = [1, 0, 1]$ as before, and the frequencies of a term t were $[2, 5, 7]^T$ in the respective versions. Then, the combined term frequency $TF_{burst}(t, d) = 1 \cdot 2 + 1/2^{\beta} \cdot 5 + (1/3^{\beta} + 1) \cdot 7$, where the third term in the sum is "boosted" by the burst in revision 3.

Having described the general burst weighting model, we now turn to the task of actually detecting the burst events.

3.2.3 Edit History Burst Detection

Documents evolve at different rates and may exhibit a variety of editing activity patterns (as captured by the revision history). For example, as news events happen, some documents may have to be updated to reflect the change in the real world. Other documents may be steadily updated by editors providing more detail or emphasizing certain topics

over others. Some of these changes are incremental and gradual, which leaves the article content relatively stable. However, some of the most important or drastic changes are reflected as "bursts" in the content or revision history (e.g., in cases where a real world event requires significant change to a document). Thus, the change in the content of a document or edit activity divides the document into natural local and global episodes that correspond to the burst. We define an edit history "burst" as either intense editing activity or dramatic content change within a time interval. Thus, we propose *content-based* and *activity-based* burst detection algorithms, and a hybrid *combined* algorithm, as described in the rest of the section.

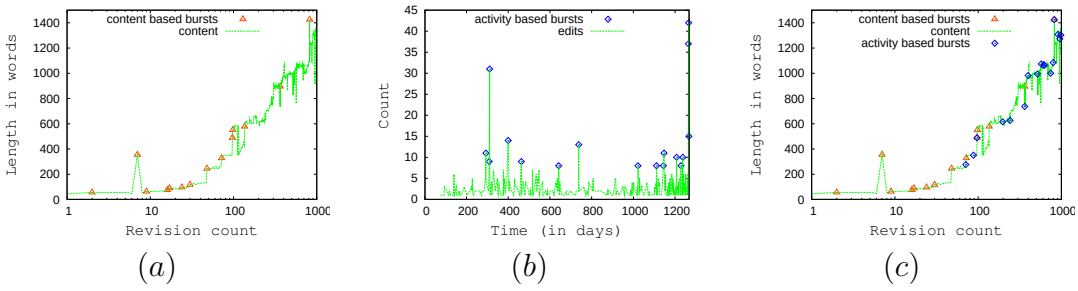

Figure 3-1: Applying the content-based (a), activity-based (b) and combined (c) burst detection methods to the Wikipedia page "Avatar".

Content-based burst detection We consider the relative content change one of the important features signaling potential bursts. The series of revisions $V = v_1, v_2, ..., v_n$ for document d are ordered by the time they appear in the revision history. For a particular pair of revisions (v_{j-1}, v_j) in this revision sequence, if the amount of content change in this interval is above a threshold α, then we consider j to be the end of a content-based burst event $Burst_c$. More formally,

$$Burst_c(v_j) = \begin{cases} 1 & \text{if } \frac{|v_j| - |v_{j-1}|}{|v_{j-1}|} > \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

where $|v_j|$ is content length for the j-th revision. The value of threshold α is important: it should not be set too high, as it may miss potential bursts, or too low, as it would cause many false bursts. After development experiments, we set $\alpha = 0.1$ for all subsequent experiments.

Activity-based burst detection To model bursts caused by intense editing activity during a certain time period, we consider the edit count in that time period as an important measure signalling bursts for a particular document. That is, we divide the revision history V of a particular document d into *episodes*, where the duration of each episode is Δt; then, an episode is considered *bursty* if the edit count for the episode exceeds "normal" amount of edit activity during the document lifetime. Then, the last revision in that *bursty* episode is selected as the end of the *burst*. More formally, a bursty episode $Burst_a$ is detected as:

$$Burst_a(ep_j) = \begin{cases} 1 & \text{if } |ep_j| > \mu + \sigma, \\ 0 & \text{otherwise.} \end{cases}$$

where μ indicates the average number of edits within an episode and σ is the standard deviation across all the episodes in the document history. For our experiments, we set the episode length to be one day.

Combined burst detection Both content-based and activity-based burst detection methods are informative as they capture significant changes in a document. Thus, we combine the two sources of information (content change and the editing activity level). In our experiments we use the simplest combination method of taking the union of the content-based and activity-based bursts. Specifically, the final set of bursts is computed as:

$$Burst(v_j) = \begin{cases} 1 & \text{if } Burst_c(v_j) + Burst_a(v_j) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

The results of different burst detection strategies above to the Wikipedia page titled "Avatar" are illustrated in Figure 3-1. Figure 3-1(a) shows the result of applying the content-based algorithm, Figure 3-1(b) illustrates the result of the activity-based algorithm, and Figure 3-1(c) shows the result of combining the two methods. As one can see, the combined model is more comprehensive as it captures both types of significant events in the “life” of a document.

Recall, that in the RHA burst model, $Burst(v_j)$ will be used as the j-th entry of burst indicator vector U. This completes the description of the RHA revision history analysis method, and we now turn to incorporating RHA into retrieval models.

3.2.4 Incorporating Word Dynamics in Retrieval Models

This section describes how RHA can be incorporated into two state-of-the-art IR models, namely, BM25 and statistical language models.

3.2.5 RHA in BM25

We now introduce the way we integrate both the global model and the burst model of RHA into the Okapi BM25 ranking function. The original Okapi BM25 ranking function is defined as:

$$S(Q,d) = \sum_{t \in Q} IDF(t) \cdot \frac{TF(t,d) \cdot (k_1 + 1)}{TF(t,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{3.5}$$

where $TF(t,d)$ represents term frequency for term t in document d. Inverted document frequency $IDF(t)$ for term t is calculated as:

$$IDF(t) = log \frac{N - n(t) + 0.5}{n(t) + 0.5} \tag{3.6}$$

where N is the number of documents in the collection, $N = |D|$, and $n(t)$ is number of documents containing term t.

BM25+RHA: We now formally define the modified BM25 model, *BM25+RHA*, that incorporates the RHA term weighting. The main change is that we replace the term frequency in BM25 with the *modified* term frequency TF_{RHA}, which is the mixture of the global and the burst models as well as the standard term frequency computed from the latest revision of the document. Specifically, TF_{RHA} is computed as:

$$TF_{RHA}(t,d) = \lambda_1 TF_{global}(t,d) + \lambda_2 TF_{burst}(t,d) + \lambda_3 TF(t,d) \tag{3.7}$$

where $TF_{global}(t,d)$ is defined in Equation 3.1, and $TF_{burst}(t,d)$ is defined in Equation 3.2. The final TF_{RHA} value is thus a linear combination of the global and burst components, weighted so that $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Using the modified term frequency TF_{RHA}, the BM25 ranking function with RHA term weighting is redefined as:

$$S_{RHA}(Q,d) = \sum_{t \in Q} IDF(t) \cdot \frac{TF_{RHA}(t,d) \cdot (k_1 + 1)}{TF_{RHA}(t,d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avgdl})} \tag{3.8}$$

The BM25 parameters are set to $k_1 = 1, b = 0.5$, which are the default values in Lemur Toolkit. While these parameters can also be optimized for the RHA modification, we decided to keep the standard Lemur parameters to make our results easier to replicate.

Extending to Multiple Field BM25 Model: Recently, a fielded variant of BM25, called BM25F, has been demonstrated to improve the performance of the BM25 model, by separately weighting the contribution of terms from the different fields in the document [73]. RHA can be naturally incorporated into BM25F by separately computing the TF_{RHA} values for each field of the document (without any additional changes to the

above method). Our preliminary experiments with BM25F+RHA model appear promising, and will be further explored in future work.

3.2.6 RHA for Statistical Language Models

We now show how RHA can be integrated into the language modeling approach for document ranking. Let D be the collection, $P(t|d)$ be the conditional probability of term t being generated by document d, and $P(t|Q)$ be the probability of term t being generated by query Q. We apply Kullback-Leibler divergence as the ranking function, following Zhai and Lafferty [51]. To score a document d w.r.t to a given query Q, we estimate the query language model and document language model, then score the document as follows:

$$S(Q,d) = D(Q||d) = \sum_{t \in V} P(t|Q) \log \frac{P(t|Q)}{P(t|d)} \tag{3.9}$$

where V is the set of all words in the vocabulary. Thus, the main task of the ranking is to estimate conditional query term probability $P(t|Q)$ and document term probability $P(t|d)$. Generally the document language model estimated with some form of smoothing, with Dirichlet prior smoothing has been showed to be one of the most effective smoothing methods. With Dirichlet prior smoothing, $P(t|d)$ estimated as:

$$P(t|d) = \frac{c(t,d) + \mu P(t|D)}{|d| + \mu} \tag{3.10}$$

where $c(t,d)$ is the count of term t in document d, μ is a smoothing parameter that is often set empirically, and $P(t|D)$ is the collection term probability which is estimated as $\frac{\sum_{d \in D} c(t,d)}{\sum_{d \in D} |d|}$. The query term probability is estimated as $P(t|Q) = \frac{c(t,Q)}{|Q|}$.

LM+RHA: We now formally define our modified LM model, *LM+RHA*. The main change to the original LM model above is that we redefine the conditional document

term probability $P(t|d)$ as $P_{RHA}(t|d)$, which in turn is computed as the linear combination of the probability derived from RHA and that from the latest version of the document:

$$P_{RHA}(t|d) = \lambda_1 P_{global}(t|d) + \lambda_2 P_{burst}(t|d) + \lambda_3 P(t|d) \tag{3.11}$$

where $P_{global}(t|d)$ is the probability of term t generated by the revision history of document d, and $P_{burst}(t|d)$ is the probability of term t generated by the bursts within the revision history of document d. Specifically, $P_{global}(t|d)$ is computed as:

$$P_{global}(t|d) = \frac{\sum_{j=1}^{n} \frac{c(t, v_j)}{j^{\alpha}}}{\sum_{t' \in d} \sum_{j=1}^{n} \frac{c(t', v_j)}{j^{\alpha}}} \tag{3.12}$$

obtained by normalizing $TF_{global}(t, d)$ with the sum of all the term frequencies of t across all the revisions of the document.

$P_{burst}(t|d)$ is defined as:

$$P_{burst}(t|d) = \frac{\sum_{j=1}^{m} \sum_{k=b_j}^{n} \frac{c(t, v_k)}{(k - b_j + 1)^{\beta}}}{\sum_{t' \in d} \sum_{j=1}^{m} \sum_{k=b_j}^{n} \frac{c(t', v_k)}{(k - b_j + 1)^{\beta}}} \tag{3.13}$$

obtained by by normalizing $TF_{burst}(t, d)$ with the sum of burst weights across all bursts in the document edit history.

Finally, the third term of Equation 3.11, $P(t|d)$, describes the probability of a term t generated by the latest version of the document d with Dirichlet prior smoothing, computed using Equation 3.10. As before, λ_1, λ_2, and λ_3 are tunable parameters that are scaled so that $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Having described how RHA can be incorporated into two example retrieval models, we now turn to the specifics of how RHA can be incorporated into a working retrieval system.

3.3 Experiments

We now evaluate RHA with the task of ranking Wikipedia pages. The following ranking methods are compared against each other:

- **BM25**: standard implementation of BM25 in Lemur, with default parameter values.
- **BM25+RHA**: Our extension of BM25 by incorporating the RHA term frequency.
- **LM**: standard implementation of the unigram language model implemented in Lemur.
- **LM+RHA**: Our extension of LM by incorporating RHA term frequency

For our study we used two different sets of query benchmarks. The first is the collection of topics and relevance judgments from the INEX 2009 Ad Hoc track evaluation [48]. The second is a set of TREC ad-hoc queries, with relevance judgments created manually for this study by volunteers.

3.3.1 Results on the INEX data

We now simulate a more realistic retrieval setting where the tuning of the parameters is performed on a separate training set to avoid overfitting. Specifically, we perform 5-fold cross validation separately on both the INEX 2008 and INEX 2009 datasets. Table 3.1 and Table 3.2 report cross-validation results on INEX 2008 and INEX 2009 query set. As the tables show, RHA consistently outperforms baseline retrieval methods.

In INEX 2008 queries, LM+RHA outperformed the baseline LM model with 8.7% relative improvement on the bpref metric and 4.9% improvement on the MAP metric. Interestingly, the improvement for the INEX 2009 is not as large as it is for INEX 2008 queries, but is still significant on bpref and R-Precision metrics. We conjecture that the

effects could be explained by the differences in the queries between the two datasets, as we analyze in more detail at the end of this section. But, in general, identifying what kind of queries could benefit from RHA is an interesting future research direction.

Model	*bpref*	*MAP*	*R-Precision*
BM25	0.307	0.281	0.324
BM25+RHA	0.312 (+1.6%)	0.291 (+3.6%)	0.320 (-1.2%)
LM	0.311	0.284	0.330
LM+RHA	0.338 (+8.7%)	0.298 (+4.9%)	0.332 (-0.6%)

Table 3.1: Retrieval performance improvements when incorporating RHA into BM25 and LM models (INEX 2008 query set with 5-fold cross validation).

Model	*bpref*	*MAP*	*R-Precision*
BM25	0.354	0.354	0.314
BM25+RHA	0.363 (+2.54%)	0.348 (-1.7%)	0.333 (+6.1%)
LM	0.357	0.370	0.348
LM+RHA	0.366 (+2.52%)	0.375 (+1.35%)	0.352 (+1.15%)

Table 3.2: Retrieval performance improvements when incorporating RHA into BM25 and LM models (INEX 2009 query set with 5-fold cross validation).

3.3.2 Results on the TREC data

Table 3.3 reports the performance results for BM25 and LM retrieval models, with and without the RHA modifications. The improvement due to incorporating RHA into the retrieval models for this benchmark are promising. This is especially true for BM25+RHA model that exhibits 3.65% relative improvement on MAP, 3.47% improvement on NDCG, and 4.39% improvement on bpref compared to the baseline BM25 model (all improvements are statistically significant with $p \leq 0.01$. These results were

obtained on a different dataset (from the INEX dataset used for tuning), without re-tuning any parameters. Therefore, these results indicate that RHA is a general and effective method for enhancing IR ranking models.

Model	*MAP*	*NDCG*	*bpref*
BM25	0.548	0.634	0.524
BM25+RHA	0.568‡ (+3.65%)	0.656‡ (+3.47%)	0.547‡ (+4.39%)
LM	0.556	0.645	0.527
LM+RHA	0.567 (+1.98%)	0.653 (+1.24%)	0.532 (+0.95%)

Table 3.3: Retrieval performance improvements for TREC queries when incorporating RHA into BM25 and LM models, ‡ indicates significant differences at the 0.01 p value using two-tailed paired t-test.

3.4 Word Frequency Change in Collaboratively Generated Content

In this section, we study how word frequency change under the influence of external events, which may suggest what types of words tend to have increased frequencies over time. When newsworthy events happen, pages in collaboratively generated content, such as Wikipedia, are often revised to update the status of the corresponding topics. In fact, incorporating information about the new events is one of the key driving forces for the continued evolution of Wikipedia pages [17] [31]. Some pages in Wikipedia are even specifically dedicated to reporting the updates for popular events. Figure 3-2 shows an example of how major events could affect the content of a Wikipedia page. In this example, a news story about a merger between two large companies triggers an update to the corresponding Wikipedia pages, where the key fact about the event is incorporated into the leading paragraph of the Wikipedia page. In this section, we investigate what

types of words tend to have increased frequency in the new revision after the related events. The results could suggest ways of modeling the word importance and weights in Wikipedia pages.

Figure 3-2: A Wikipedia page before and after a related news event.

We emphasize that modeling content change in response to an event is a distinct problem from single- or multi-document summarization. Our pilot study shows that even human-generated event summaries exhibit little overlap with the event-related word frequency change. One explanation is that different pages tend to focus on different aspects of an event. The task we address differs from summarization in that we aim to distinguish which parts of the event description will be chosen to include into which page. For example, to incorporate the event content and make it coherent with the original CGC page content, the editors may have to introduce the context and background of the event, to change outdated information about the event, and to link the content to other concepts in Wikipedia.

In this work, we investigate event-driven content change in CGC (focusing on Wikipedia), and characterize how events drive the content change on word frequencies. Our preliminary results provide insights for building a predictive model of event-driven content change in CGC. To the best of our knowledge, this is the first study of this problem, and our findings could shed light on understanding how news events influence content change and formation of new knowledge in collaboratively generated content.

3.4.1 Definition

By Event-driven content change we mean the natural process of Wikipedia editing to incorporate event content into the Wikipedia page. In our settings, it means whether the frequency of the event-related phrase are changed.

Event-related phrases: In this study, we define event-related phrases as any of the subtrees from context-free phrase structure grammar representation extracted from event articles (using the Stanford parser [24]). The heuristic is that phrases used in the event news articles are more essential to describe the event and have higher chance to be incorporated into Wikipedia pages than others. Notice that this definition could result in phrases ranging from a single word to whole sentence. In this study, we limit the phrase length to at most 3 words.

Three components are involved in event-driven content change:

- The original content of the Wikipedia pages.
- Events: Major events are usually newsworthy and news articles are often well written and have rich syntactic information. In this study, we represent major events with news articles. We follow the definition of events in Topic Tracking and Detection [4]: "a particular thing that happens at a specific time and place, along with all necessary preconditions and unavoidable consequences".
- The relevance between Wikipedia pages and events.

3.4.2 Dataset

To create the dataset for our study, we use a valuable, and we believe under-utilized resource – a specialized portal in Wikipedia called "current events". The editors of this page select the most important major events from external news web sites and summarize the events with references to the relevant Wikipedia pages and news websites.

This resource allows us to explicitly link a set of content changes in Wikipedia to the corresponding news event.

Figure 3-3: An example entry from "Current events" page, illustrating a summary for the event in Figure 1.

Figure 3-3 shows an example of events in "Current events" page. The setting of the page provides information of all three components in event-driven content change. In this study, we collected all events in 2013 from "Current events" page and their corresponding news articles. To obtain the change in the related Wikipedia pages, we crawl the page snapshot created before the event date, and the snapshot of the page one day after the event date, which leaves one day for editors to update the page with event-related information.

Number of events	1281
Number of Relevant Wikipedia Pages	4496
Average Paragraph Count in Wikipedia Pages	91.8
Number of Relevant Wikipedia Pages Per Event	3.5
Number of Event Categories	31

Table 3.4: Description of Collected Wikipedia "Current event" Data.

3.4.3 Analysis of Word Frequency Change

We now consider which features could potentially help predict event-driven word frequency change on phrase level. We develop and investigate the following features ex-

Features	Info Gain
PhraseCountInWiki	0.030
PhraseCountInNews	0.024
PhraseCountInWikiFirstPara	0.019
NounPhraseInWikiFirstPara	0.018
NounPhraseInNews	0.017

Table 3.5: Features with top Info Gain.

tracted only from Wikipedia pages and events:

- Features from Wikipedia pages before events:
  - Frequencies.
  - Locations of the phrases (e.g., in title, first paragraph etc.).
  - Part-of-speech tags if available.
  - Phrase structure grammar tags in the first paragraph if available.
- Features from news articles: similar with features from Wikipedia pages.

The feature contributions are evaluated by computing their Information Gain against the ground truth labels. Again, the ground truth labels indicate whether phrase frequency were changed due to the events. Table 3.5 lists the top 5 features ranked by their Information Gain value.

The findings are: (1) The most promising feature is the count of event-related phrases in Wikipedia pages. In other words, frequent phrases would be more "frequent". This finding aligns with the assumption in [3], where they claimed important words would appear in early revisions of Wikipedia pages. (2) The location of phrases matters. Phrases that are in the first paragraph of Wikipedia pages tend to have increased frequencies. (3) Surprisingly, phrases in news titles are not effective features, implying that different Wikipedia pages favor different aspect of the events. Words used in news titles are not necessarily critical to Wikipedia pages.

Between revisions, frequent words in the older revision have higher probability to become more frequent in the newer one. When we extend this result to the whole revision history of Wikipedia pages, it means frequent words would be more frequent over time. If we consider "frequent" as a signal of importance, then the finding suggests count of important words tend to increase all the time. RHA [3] model works in the same way: weighting words by their frequency over time.

3.5 Summary

We introduced a novel term weighting scheme that uses Revision History Analysis (RHA) of the document edit history. Unlikely previous models, RHA directly captures the document authoring process when available, and is particularly valuable for collaboratively generated content, notably Wikipedia documents. RHA can be naturally incorporated into state of the art retrieval models, as we demonstrate by showing consistent improvements that RHA enables for BM25 and LM retrieval models. Other potential applications of RHA include document classification, clustering, and feature selection – as all of these tasks make use of the term frequency information. Additionally, we propose to model and predict word dynamics to events. The analysis shows that frequent words in early revisions tend to have increased frequency after the events. This finding validates the design of our RHA model and could imply other applications which use word frequencies.

Chapter 4

Modeling Topic Dynamics in Microblogging Content

Topics inferred from the content inherit dynamics from the content evolution. Topics in personal MBC, such as Tweet streams from an author, can be quite dynamic. Regular Twitter users rarely post Tweets about the same topics over and over. Instead, the topics generated by a user often transit from one to another. We propose to investigate topic dynamics in microblog streams. The goal is to learn how topics transition from one to another in the post stream of a user and predict the future topics in his stream according to the learned historical transition patterns.

4.1 Topic Dynamics in Tweet Streams

Consider a Tweet stream of an author. The timestamp of each Tweet determines the order of it along the timeline. Since Tweets reflect activities or status of the author, the temporal order of tweets reflects the time dimension of the author's behavioral patterns. Thus the temporal sequence of a Twitter stream is a factor connecting tweet content and one's real life activities. Users' tweets include a variety of topics and rich information,

such as breaking news, their comments on popular events, daily life events and social interaction. Obviously, the topics of tweet streams will not be static, but change over time. In other words, users tend to tweet about different topics instead of simply repeat previous tweets. Thus, to better model the dynamic semantics of tweet streams, we need a temporal-sensitive model that can capture the changing pattern among topics. The implications of better modeling topic dynamics reach far beyond Twitter, as most social textual data are naturally sequenced by time. Better understanding and modeling of of the temporal dynamics of social content can not only benefit these applications, but provide powerful analytical tools for researchers and analysts.

4.2 Modeling Topic Transitions

We propose *Temporal Latent Dirichlet Allocation* (TM-LDA) to model topic transitions in temporally-sequenced documents. In the case of Tweeter streams, we claim that topic transitions of an author's tweets follow certain cause-effect rules or social behavioral patterns. For example, people tend to talk about the topic "Drink" after "Food", which implies a certain dietary and social manner. In some cities, users complain about "Traffic" mostly after they tweet about "Places", which reflects poor traffic condition in those areas. Understanding these topic transition rules is meaningful in three ways:

- Dynamically predicting future trends of tweet stream based on the historical tweets
- A tool to provide analysts a more in-depth view of causal relationships among social phenomena. For instance, the factors or topics leading to "Traffic" will be interesting to the traffic department.
- Providing a signal of unusual events when topics fail to follow common transition rules.

TM-LDA is designed to learn the topic transition parameters from historical temporally-sequenced documents to predict future topic distributions of new documents over time. TM-LDA takes pairs of consecutive documents as input and finds the optimal transition parameters, which minimize the least squares error between predicted topic distribution and the actual topic distribution of the new tweets. Additionally, transition parameters among topics can vary over time because of the changing popularity of certain topics and external events. To adaptively update the transition parameters as the new tweets stream in, we propose an efficient algorithm which can adjust the transition parameters by appending new consecutive tweet pairs into the system and deleting outdated tweet pairs.

4.2.1 Temporal Latent Dirichlet Allocation (TM-LDA)

We design TM-LDA as a system which generates topic distributions of new documents by taking previous documents as input. More precisely, if we define the space of topic distribution as $X = \{x \in \mathbb{R}_+^n : ||x||_1 = 1\}$, TM-LDA can be considered as a function $f : X \to X$. Notice that n is the dimension of the space X, in other words, n is the number of topics; $||\cdot||_1$ is the ℓ^1 norm of vector x. Given the topic distribution vector of a historical document x, the estimated topic distribution of a new document $\hat{y}$ is given by $\hat{y} = f(x)$. Once we know the real topic distribution of the new document y, the prediction error of the TM-LDA system would be:

$$err_f = ||\hat{y} - y||_2^2 = ||f(x) - y||_2^2.$$

Function err_f uses the ℓ^2 norm to measure the prediction error because the minimization of err_f can thus be reduced to a least squares problem, which can be efficiently solved. The training stage of TM-LDA is to find the function f which minimizes err_f.

In our system settings, x and y are topic distribution vectors of two consecutive

tweets, where x represents the "old" tweet, and y corresponds to the "new" tweet. TM-LDA predicts the topic distribution of y by taking historical tweet x as input and applies function f on it to obtain $\hat{y}$. Therefore the prediction error of TM-LDA is the difference between $\hat{y}$ and y.

In our work, TM-LDA is modeled as a non-linear mapping:

$$f(x) = \frac{xT}{||xT||_1}, \tag{4.1}$$

where x is a row vector, $T \in \mathbb{R}^{n \times n}$. The product of x and T is also a row vector, which is the estimated new topic weighting vector (before normalization). After xT is normalized by its ℓ^1 norm, it becomes a topic distribution vector.

4.2.2 Error Function of TM-LDA

Function (4.1) defines the prediction function for a single document or tweet x. The error function is therefore:

$$err_f = \left|\left| \frac{xT}{||xT||_1} - y \right|\right|_2^2. \tag{4.2}$$

Intuitively, this function measures the prediction error for a single pair of documents, x and y, where x represents the "old" document and y is the "new" document. Now we generalize it and define the error function for a collection of documents. Suppose we have a collection of sequenced documents D, where the number of documents is $|D| = m + 1$; the topic distribution of the i-th document is d_i, where i indicates the temporal order of d_i. Next, we construct two matrices $D^{(1,m)}$ and $D^{(2,m+1)}$ as follows:

$$D^{(1,m)} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix}, \quad D^{(2,m+1)} = \begin{bmatrix} d_2 \\ d_3 \\ \vdots \\ d_{m+1} \end{bmatrix}.$$

Notice that both $D^{(1,m)}$ and $D^{(2,m+1)}$ are $m \times n$ matrices. The i-th rows of these two matrices are d_i and d_{i+1}, and they are sequentially adjacent in the collection D. In other words, $D^{(1,m)}$ represents the topic distribution matrix of "old" documents and $D^{(2,m+1)}$ is the matrix of "new" documents. According to the error function for a single document pair (Function (4.2)), the prediction error for the sequenced document collection D is defined as:

$$err_f = ||LD^{(1,m)}T - D^{(2,m+1)}||_F^2. \tag{4.3}$$

where $||\cdot||_F$ is the Frobenius matrix norm. L is a $m \times m$ diagonal matrix which normalizes each row of $D^{(1,m)}T$. The i-th diagonal entry of L is the reciprocal of the ℓ^1-norm of the i-th row in $D^{(1,m)}T$:

$$L = \begin{bmatrix} \frac{1}{||d_1T||_1} & & & \\ & \frac{1}{||d_2T||_1} & & \\ & & \ddots & \\ & & & \frac{1}{||d_mT||_1} \end{bmatrix}.$$

4.2.3 Iterative Minimization of the Error Function

The function err_f is a non-linear function. Numerical experiments show that function err_f is convex, which suggests using iterative methods to approach the optimal T that minimizes err_f. Each iteration updates the solution T as below:

$$T^{(j)} = (L^{(j-1)} D^{(1,m)})^{\dagger} D^{(2,m+1)},$$

where

$$L^{(j-1)} = \begin{bmatrix} \frac{1}{||d_1 T^{(j-1)}||_1} & & \\ & \ddots & \\ & & \frac{1}{||d_m T^{(j-1)}||_1} \end{bmatrix}.$$

Such iterative method can be initialized by

$$T^{(0)} = D^{(1,m)\dagger} D^{(2,m+1)},$$

where $D^{(1,m)\dagger}$ is the pseudo-inverse of $D^{(1,m)}$.

4.2.4 Direct Minimization of the Error Function

Iterative methods may be slow to converge and only give an approximate solution. Ideally, we would like to have a direct solution procedure for TM-LDA which could be efficiently and accurately implemented. By noticing an important property of the TM-LDA error function, we use Theorem 1 to derive a least squares characterization of the TM-LDA solution and to provide the explicit form of the exact solution.

Theorem 1. *Let* $\mathbf{e}$ *denote the* $n \times 1$ *matrix of all ones. For any* $A \in \mathbb{R}_{+}^{m \times n}$ *and* $B \in \mathbb{R}_{+}^{m \times n}$ *such that* $A\mathbf{e} = \mathbf{e}$ *and* $B\mathbf{e} = \mathbf{e}$*, it holds*

$$AA^{\dagger}B\mathbf{e} = \mathbf{e},$$

where $A^{\dagger}$ *is the pseudo-inverse of* A*.*

Proof. Because $B\mathbf{e} = \mathbf{e}$,

$$AA^{\dagger}B\mathbf{e} = AA^{\dagger}\mathbf{e}.$$

$AA^{\dagger}\mathbf{e}$ is the orthogonal projection of $\mathbf{e}$ onto $Range(A)$. Since $A\mathbf{e} = \mathbf{e}$, $\mathbf{e} \in Range(A)$. Therefore $AA^{\dagger}\mathbf{e} = \mathbf{e}$. □

The matrices $D^{(1,m-1)}$ and $D^{(2,m)}$ satisfy the properties $D^{(1,m-1)}\mathbf{e} = \mathbf{e}$ and $D^{(2,m)}\mathbf{e} = \mathbf{e}$ since each row of these two matrices is a topic distribution vector of a document and the row sum is naturally 1. By adapting the result of Theorem 1 to TM-LDA, we obtain the following result:

$$D^{(1,m)}T^{(0)}\mathbf{e} = D^{(1,m)}D^{(1,m)\dagger}D^{(2,m+1)}\mathbf{e} = \mathbf{e}.$$

In other words, $\|d_i T^{(0)}\|_1 = 1$ for any $i \in \{1, 2, \ldots, m\}$. Therefore $L^{(0)} = I$, the $m \times m$ identity matrix. Hence, $T^{(1)}$ can be written as

$$T^{(1)} = (L^{(0)}D^{(1,m)})^{\dagger}D^{(2,m+1)} = T^{(0)}.$$

This indicates that

$$T = D^{(1,m)\dagger}D^{(2,m+1)}$$

gives the optimal solution for minimizing err_f. Hence, computing the TM-LDA solution amounts to solving a matrix least squares problem:

$$\min_{T} ||D^{(1,m)}T - D^{(2,m+1)}||_F^2.$$

4.3 TM-LDA for Twitter Stream

A Twitter stream of an author consists of temporally sequenced Tweets. After we train LDA on the collection of Tweets, the topic distribution vector of each Tweet is obtained. We can therefore construct the matrices $D^{(1,m)}$ and $D^{(2,m+1)}$. Suppose we collect 20

consecutive Tweets per unique user and the number of unique users is p, then the training stage of TM-LDA on such Twitter stream dataset is illustrated in Figure 4-1. The left matrix is $D^{(1,m)}$ and the right matrix is $D^{(2,m+1)}$, where $m = 19 \times p$ in this case. For each user, 20 consecutive Tweets makes 19 tweet pairs, they are (tweet 1, tweet 2), (tweet 2, tweet 3), ..., (tweet 19, tweet 20). Each Tweet pair is one training sample and forms one row of matrix $D^{(1,m)}$ and $D^{(2,m+1)}$. By multiplying the "old" Tweet matrix $D^{(1,m-1)}$ with the transition parameter matrix T, the predicted topic distribution of "new" Tweets is obtained.

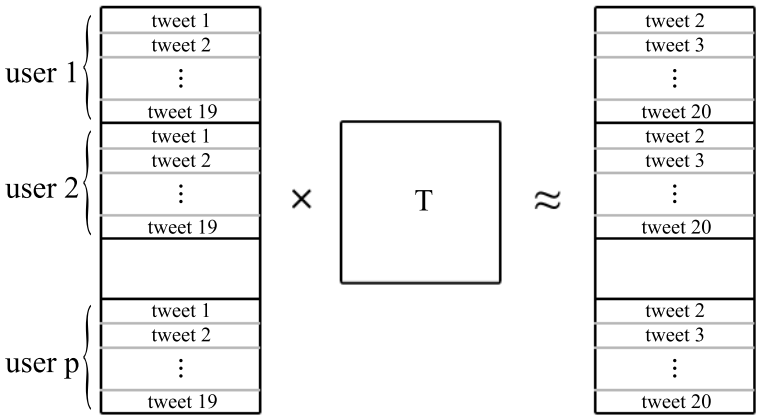

Figure 4-1: Constructing TM-LDA for tweets.

To simplify the notations, let $A = D^{(1,m)}$ and $B = D^{(2,m+1)}$. According to Theorem 1, TM-LDA is reduced to the following problem:

$$\min_{T} ||AT - B||_F^2. \tag{4.4}$$

Again, A is the topic distribution matrix of "old" tweets and B is the topic distribution matrix of "new" tweets. The training phase of TM-LDA becomes a least squares problem. When the condition number of A, $\kappa(A)$, is small and the system is overdetermined, we can effectively obtain T as

$$T = (A'A)^{-1}A'B, \tag{4.5}$$

where A' denotes the transpose of A. In practice, multiplication by $(A'A)^{-1}$ is accomplished by Cholesky factorization of $A'A$ followed by forward and backward substitutions.

4.4 Updating Transition Parameters

Not only the topics of a user's twitter stream will change, but the transition weights from one topic to another also vary over time. Both the changing popularity of certain topics and external events will affect the transition parameters of related topics. In other words, the transition parameters have to be updated and adjusted by taking recently generated tweets as training samples. One way to solve this updating problem is to compute the transition parameter matrix every time new Tweets come in. However, re-computing transition parameters may result in lower efficiency and a less smoothly changing parameter adjustment process. In this section, we will show an efficient algorithm which can gradually and smoothly adjust transition parameters as the new tweets are generated with much less computation than re-computing TM-LDA.

We now introduce the algorithm to perform updating transition parameter matrix T. Suppose we append k rows of new tweet pairs, U_k and V_k, to the bottom of A and B and form $\hat{A}$ and $\hat{B}$ as below:

$$\hat{A} = \begin{bmatrix} A \\ U_k \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B \\ V_k \end{bmatrix}.$$

Then according to Equation (4.5), the new transition parameter matrix, $\hat{T}$ is:

$$\hat{T} = (\hat{A}'\hat{A})^{-1}\hat{A}'\hat{B}.$$

We apply the Sherman-Morrison-Woodbury formula [35] to $(\hat{A}'\hat{A})^{-1}$ and obtain the

following result:

$$(\hat{A}'\hat{A})^{-1} = (A'A + U_k^{'}U_k)^{-1}$$

$$= (A'A)^{-1} - (A'A)^{-1}U_k^{'}(I + U_k(A'A)^{-1}U_k^{'})^{-1}U_k(A'A)^{-1}.$$

Let $C = (A'A)^{-1}U_k^{'}$, then the updated transition parameter matrix $\hat{T}$ is:

$$\hat{T} = (\hat{A}'\hat{A})^{-1}(A'B + U_k^{'}V_k)$$

$$= T + CV_k - C(I + U_kC)^{-1}C'(A'B + U_k^{'}V_k). \tag{4.6}$$

Notice that $A'A$ and $A'B$ have been computed and stored when computing T. In other words, to compute $\hat{T}$, we just need $U_k^{'}V_k$ and C. The only possibly expensive part is to obtain $(I + U_kC)^{-1}C'$, which requires $O(k^3)$ at most [35]. The remaining components of computing $\hat{T}$ have the complexity of $O(k)$, and even less when U_k and V_k are sparse. Therefore the overall cost for updating the transition parameter matrix is $O(k^3)$ or less.

4.5 Experiments

In this section, TM-LDA is evaluated against large-scale Twitter stream data. By measuring perplexity, we show that TM-LDA significantly outperforms static topic models on predicting actual word distributions of future tweets. Additionally, the efficiency of the algorithm for updating transition weights is also assessed.

4.5.1 Dataset

To validate TM-LDA, we collect Tweets from more than 260,000 public user accounts over one month. The public user accounts are selected from the TREC 2011 microblog

track[1] and we only keep the users with valid geo-location information. A list of 89 candidate cities are generated by taking the union of top 50 U.S. cities (in population) and the capital cities of the 50 U.S. states. After that, the users whose claimed geo-locations are one of the candidate cities will be selected.

All selected user accounts are tracked daily and they generate an average of around 1.1 million new tweets per day. However, tweets are usually short and informal which makes the quality of tweets vary a lot from each other. To control the quality of tweets, we first filter out stopwords and the words with less than 5 occurrences in our dataset, and then keep the tweets with more than 3 terms left. In this way, one third of the raw tweets are filtered, resulting in more than 20 million "high quality" tweets.

Dates	From 12-15-2011 To 1-15-2012
Number of Raw Tweets	34,150,390
Number of Valid Tweets	23,096,894
Average Length of Valid Tweets (words)	5.12
Number of Users	264,628
Number of Cities	89
Number of Valid Tweet Pair	13,273,707

Table 4.1: Description of Twitter Stream Data.

4.5.2 Using Perplexity as Evaluation Metric

TM-LDA is designed to predict the topic distribution of future tweets based on historical tweets. Therefore we employ the measurement of *Perplexity* to evaluate TM-LDA against the actual word occurrences in future tweets. Usually, perplexity is used to measure how well a language model fits the word distribution of a corpus. It is defined as:

$$Perplexity_l = 2^{-\sum_{i=1}^{N} \log_2 p_l(x_i)}. \tag{4.7}$$

[1]http://trec.nist.gov/data/tweets/

Formula (4.7) measures the perplexity of the language model l, where $p_l(x_i)$ is the probability of the occurrence of word x_i estimated by the language model l and N is the number of words in the document. Intuitively, if the language model yields higher probability for the occurrences of words in the document than words that are not in the document, the language model is more accurate and the perplexity will be lower.

4.5.3 Predicting Future Tweets

TM-LDA predicts the topic distribution of future tweets by taking the "previous" tweets as input (Formula (4.1)). Basically, TM-LDA will multiply the topic distribution vector by the transition parameter matrix and normalize it to form the topic distribution of the "future" tweet. There are two key components in this process: (1) the transition parameter matrix, and (2) the topic distribution of "previous" tweets.

The transition parameter matrix is trained according to the algorithm introduced in Section 4.2.4. In practice, TM-LDA will use 7-day (one week) historical tweets to train the transition parameter matrix, and then predict the tweets generated on the 8th day. For example, if we want to predict the tweets on the date Dec. 22, 2011, we will collect all the tweets generated from Dec. 15, 2011 to Dec. 21, 2011 and train LDA on this one-week tweet collection to obtain the topic distribution vectors for each single tweet. During the training of LDA, each tweet is treated as a document and the number of topics is set to 200. After that, we build two topic distribution matrices, "old" tweet matrix and "future" tweet matrix, as in Figure 1 and compute the transition parameter matrix according to Formula (4.5).

For the tweets generated on the 8th day (which we want to predict), we cannot have their topic distributions from LDA directly. Figure 2 shows the circumstances: LDA is trained on one-week tweets but not on the tweets a and b, which means we need to map them to the topics through the results of LDA. The topic distribution of "previous" tweets a is inferred from the LDA model. Given the words appeared in the tweet t, the

topic distribution is inferred as:

$$p(z|t) = \sum_{w} p(z|w)p(w|t) = \sum_{w} \frac{p(w|z)p(z)}{\sum_{z'} p(w|z')p(z')} p(w|t), \tag{4.8}$$

where $p(w|t)$ is the normalized frequency of word w in tweet t. Both $p(w|z)$ and $p(z)$ are the results of LDA model.

In summary, TM-LDA first trains LDA on 7-day historical tweets and compute the transition parameter matrix accordingly. Then for each new tweet generated on the 8th day, it predicts the topic distribution of the following tweet. When the actual "future" tweet b (in Figure 4-2) becomes available, we can therefore measure the perplexity of TM-LDA.

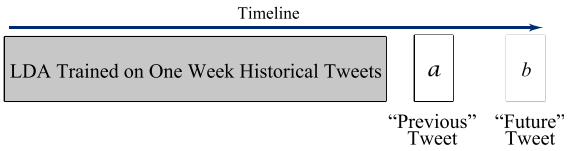

Figure 4-2: The Scheme for Predicting "Future" Tweets.

Figure 4-2 illustrates the prediction scheme of TM-LDA and other methods. They build LDA on one-week historical tweet data, and for each new tweet a, they predict the topic distribution of the "following" tweet b. Although many dynamic topic models are developed [14, 86, 49], they are mainly designed to model topic trends and dynamic word distributions over time, instead of predicting future topic distributions. Therefore, we compare TM-LDA with the following methods:

1. **Estimated Topic Distributions of "Future" Tweets**: the topic distribution of the tweet b. This is computed based on the actual words in the "future" tweets according to Formula (4.8). This system approximately reflects the optimal perplexity of LDA-based models.

2. **LDA Topic Distributions of "Future" Tweets**: the inferred topic distribution of

the tweet b. They are inferred from the LDA model which is trained on the one-week historical tweets. The inferring algorithm is introduced by Blei et al. [15]. This system knows the words appearing in the "future" tweets, so that it shows the optimal perplexity of the original LDA [15].

3. **LDA Topic Distributions of "Previous" Tweets**: the inferred topic distribution [15] of the tweet a. They are also inferred from the LDA trained on one-week historical tweets. This system uses the topic distributions of "previous" tweets as the topic distributions of the "Future" tweets. It shows the perplexity of static prediction model built on LDA.

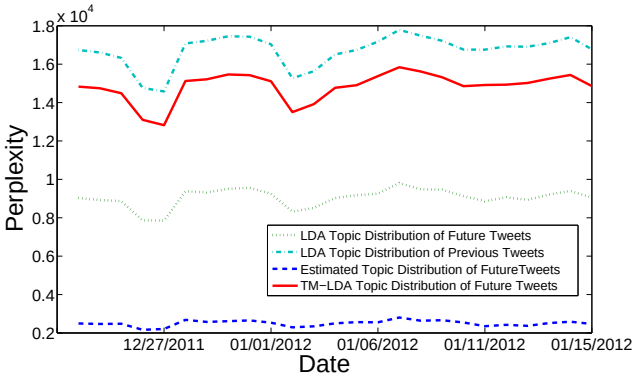

Figure 4-3: Perplexity of Different Models.

We test these 3 methods and our model, TM-LDA, on the tweets generated from 12/22/2011 to 01/15/2012. In Figure 4-3, we can see that TM-LDA consistently provides lower perplexity than the static prediction model. The improvements are statistically significant with $\alpha < 0.001$. It turns out that the performance of TM-LDA could be affected by the topic estimation of "previous" tweets, which TM-LDA uses as input arguments. One interesting fact is that tweets are easier to predict on holidays than other days. We can see that the perplexity drops on the dates of Christmas and New Year, which suggests that the topics discussed during holiday seasons are more predictable.

Also note that we use the ℓ^2 norm to define the error function for TM-LDA, which enables us to efficiently optimize it by solving a least squares problem.

4.5.4 Efficiency of Updating Transition Parameters

In Section 4.4, we introduced the Sherman-Morrison-Woodbury formula to update the transition parameter matrix. Now we turn to show the runtime complexity of this algorithm. Suppose we have computed a transition parameter matrix T based on one-week historical tweet data, which consists of more than 3 million tweet pairs. Given k new pairs of tweets, we measure the time needed to update matrix T for different values of k.

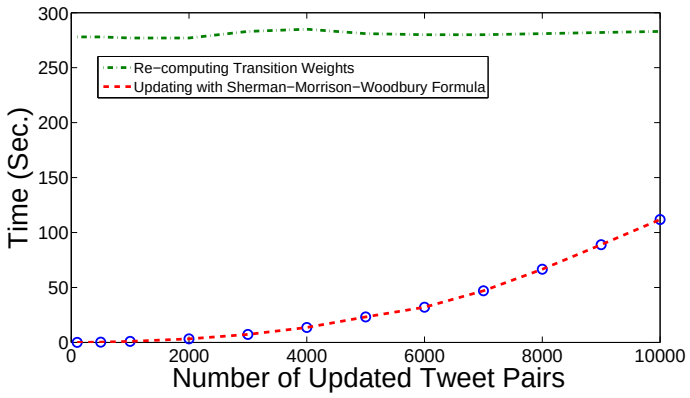

Figure 4-4: Time Complexity of Updating Transition Parameter Matrix based on One-Week Tweet Data.

We test the time complexity by running the Matlab implementation of our updating algorithm on a machine with 24 *AMD Opteron(tm) 6174* processors and 128 Gigabytes memory. Figure 4-4 shows that our updating algorithm can efficiently find T when k is not too large. Compared with re-computing the matrix T, which usually takes around 280 seconds for one-week tweet data, our updating algorithm consumes less time, while resulting in a more smoothly varying T.

4.6 Visualization and Sensemaking of Topic Transitions

TM-LDA can provide a more in-depth view of cause-effect relationships among topics and public opinion of popular events. We now turn to discuss the analytical power of TM-LDA.

4.6.1 Global Topic Transition Patterns

To show the global topic transition patterns, TM-LDA is trained on all the valid tweet pairs we've collected. The topic transition parameter matrix T has the size of 200×200 and the average transition weight of all 40000 entries in T is 0.005. We visualize the matrix T as in Figure 4-5 (a); however, this figure is not quite clear and it is challenging to locate interesting topic transition patterns. We therefore develop an algorithm to pre-select "interesting points" from the raw transition matrix, and then do a case study on those "interesting transition points".

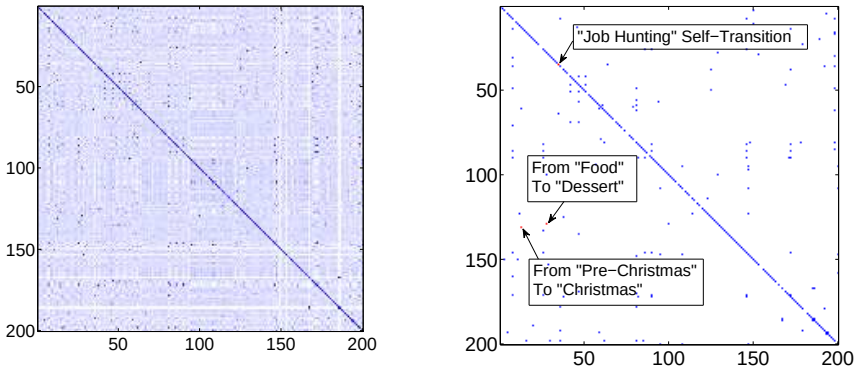

Figure 4-5: Visualization of topic transitions: (a) global topic transitions (b) interesting transition points after filtering.

In Figure 4-5 (a), it's clear that matrix T has large diagonal entries. This provides the evidence that topics of historical tweets do not randomly transit from one to another, but

	From Topic	*To Topic*	*Weight*
(1)	Job Hunting	Job Hunting	0.448
	Weather	Weather	0.304
	Reading Media	Reading Media	0.286
	Weight Loss	Weight Loss	0.282
(2)	Internet Company	Social Media	0.045
	U.S. Deficit	Presidential Election	0.044
	Food	Dessert	0.041
	Security and Crime	Military Action	0.039
(3)	Traffic and Accident	Rescue and Police	0.044
	Restaurant	Food	0.040
	Pre-Christmas	Christmas	0.032
	Startup Business	Social Media	0.030

Table 4.2: Three Kinds of Topic Transitions: (1) Self-Transition (2) Symmetric Transition (3) Non-Symmetric Transition.

follow certain statistical rules. However, the diagonal entries of T are always less than 1. Meanwhile, the empirical average value of diagonal entries in T, $\bar{t}$ is 0.095, which shows that new tweets usually do not simply repeat the topics of historical tweets. The standard deviation of non-diagonal (non-self-transition) entries, σ, is 0.003. We define the threshold to be the average plus five times the standard deviation: $Threshold = \bar{t} + 5 \times \sigma$, which is $0.005 + 0.003 \times 5 = 0.02$, as the bar of "interesting" points. After filtered by this threshold, a more clear transition pattern is obtained and shown in Figure 4-5 (b).

Figure 4-5 (b) shows three kinds of "interesting" transition points: (1) diagonal points: these points have high self-transition weights and they are the topics people tend to keep discussing about; (2) symmetric points: both t_{ij} and t_{ji} are interesting points. These topics are highly correlated and they are usually mentioned in consecutive tweets; (3) non-symmetric points: one and only one of t_{ij} and t_{ji} is an interesting point. These topics usually reflect strongly time-sensitive properties of certain events and scenarios. We rank the points in Figure 6 by their values and list the most representative ones in Table 4.2.

Table 4.2 shows the general topic transition patterns of Tweet streams. We can tell that certain topics are very popular according to their high self-transition weights, such topics include "Job hunting" and "Weight loss". The topic popularity provided by TM-LDA not only show the amount of related tweets, but also reflect the persistence of certain topics. Besides this, transition weights can also be indicators of relatedness among topics. For example, the topic "Internet company" and the topic "Social media" are very close to each other and therefore one topic could trigger users' interest in the other topic. Additionally, we can also find some "one way" transitions, which may suggest strong temporal orders or cause-effect relationships among topics. For instance, the topic "Pre-Christmas" is about the ideas and preparation of Christmas gifts; this topic always appears before the topic "Celebration of Christmas". This information is very useful not only for predicting future tweets, but for personalization systems and advertising industries.

4.6.2 Various Topic Transition Patterns by Cities

Topic transition patterns can help reveal potential social issues and identify interesting behavioral patterns in various cities. We study the transition parameter matrices over nine major cities in the United States. Empirical results show that these cities have very different topic transition weights from each other.

The topic transitions of cities are studied in two aspects: (1) for a particular topic, which topics tend to occur before this topic (Table 4.3); and (2) the topics appearing after this topic (Table 4.4). The first aspect tells us what could be the causes of a topic/problem, and the second aspect shows what is the next possible event after an activity/topic.

Table 4.3 lists the sample topic transitions of 9 cities, and it reflects the different problems and characteristics of different places. For example, the topics occurring before "Compliments" could potentially be able to please people, and the topics before

	Traffic	*Complaints*	*Compliments*
Atlanta	Airport	Smoke/Drug	Holidays
Boston	Trip	Music	Love
Chicago	Weather	Work Life	Pray
Los Angeles	Church	Break-up	Basketball
Miami	Party	Alcohol	Holidays
New York	Manhattan	Break-up	Movies
San Francisco	Japan/Sushi	Hate	Love
Seattle	Weather	Party	Planning
D.C.	Plaza	Sleep	Dress

Table 4.3: The Top Topics before "Traffic", "Complaints" and "Compliments".

"Complaints" might be related to social problems.

The result of TM-LDA can also benefit targeted analysis. In Table 4.3, we show the top topics occurring before "Traffic", which may imply the potential traffic issues in various cities. It turns out that the results align with the actual traffic conditions quite well, such as the airport in Atlanta (the busiest airport in the world), Manhattan area in New York and Japan town in San Francisco.

	Work Life	*Dining*
Atlanta	Complaint	Party
Boston	Book	Beauty
Chicago	Celebration	Weight Loss
Los Angeles	E-shopping	Beauty
Miami	Music	Shopping
New York	Social Media	Weight Loss
San Francisco	Weight Loss	Entertainment
Seattle	Job Hunting	Weight Loss
D.C.	Presidential Election	Reading Media

Table 4.4: The Top Topics after "Work Life" and "Dining".

Table 4.4 shows the topics mentioned after "Work life" and "Dining". It provides the observation of what people tend to do or discuss after work and dinner. Advertising can be more content-aware and targeted with this information. For example, the users in Los Angeles and Boston would like to talk about facial and beauty after dinner, which

suggests a better advertising strategy in these cities. More importantly, these results also imply the different behavioral patterns of cities which could help people to better understand the culture of different places.

4.7 Summary

We presented and evaluated a novel temporally-aware language model, TM-LDA, for efficiently modeling streams of social text such as a Twitter stream for an author, by modeling the topics and topic transitions that naturally arise in such data. We have shown that our method is able to more faithfully model the word distribution of a large collection of real Twitter messages compared to previous state-of-the-art methods. Furthermore, we introduced an efficient model updating algorithm for TM-LDA that dramatically reduces the training time needed to update the model, making our method appropriate for online operation. Finally, in a series of experiments, we demonstrated ways in which TM-LDA can be naturally applied for mining, analyzing, and exploring temporal patterns in Twitter data.

Chapter 5

Modeling and Analyzing Sentiment Dynamics in Microblogging Content

Sentiment analysis has been extensively studied in both traditional content and MBC. However, the research challenge of modeling sentiment dynamics is not fully explored and investigated. Sentiment in MBC changes over time as people express and amplify opinions, especially when major events are happening. Different opinions and sentiment come from users with different background, interests, beliefs, etc. It is difficult to model and understand sentiment change without truly knowing the users' characteristics.

In this work, we take two steps to model sentiment changes. First, we apply existing state-of-the-art techniques to describe mixed and aggregated sentiment dynamics during the period of major events. Second, we propose a user characteristic inference model to divide the mixed sentiment into finer grained ones, which contextualizes raw sentiment and opinion with users' attributes. The techniques developed in this chapter build the basics for deep analysis of sentiment and its dynamics in MBC, which in turn offer a great opportunity to validate theories and hypotheses in the domain of social science. Particularly, we focus on sentiment of political issues, which are one of the most popular topics in MBC.

5.1 Sentiment and Sentiment Dynamics in Twitter

People express and amplify political opinions in Microblogs such as Twitter, especially when major political decisions are made. Twitter provides a useful vehicle for capturing and tracking popular opinion on burning issues of the day. In this section, we track the changes in political sentiment related to the U.S. Supreme Court (SCOTUS) and its decisions, focusing on the key dimensions on support, emotional intensity, and polarity. Measuring changes in these sentiment dimensions could be useful for social and political scientists, policy makers, and the public. This preliminary work adapts existing sentiment analysis techniques to these new dimensions and the specifics of the corpus (Twitter). We illustrate the promise of our work with an important case study of tracking sentiment change building up to, and immediately following one recent landmark Supreme Court decision. This example illustrates how our work could help answer fundamental research questions in political science about the nature of Supreme Court power and its capacity to influence public discourse.

Political opinions are a popular topic in Microblogs. On June 26th, 2013, when the U.S. Supreme Court announced the decision on the unconstitutionality of the "Defense of Marriage Act" (DOMA), there were millions of Tweets about the users' opinions of the decision. In their Tweets, people not only voice their opinions about the issues at stake, expressing different dimensions of sentiment, such as support or opposition to the decision, or anger or happiness. Thus, simply applying traditional sentiment analysis scales such as "positive" vs. "negative" classification would not be sufficient to understand the public reaction to political decisions.

Research on mass opinion and the Supreme Court is valuable as it could shed light on the fundamental and related normative concerns about the role of constitutional review in American governance, which emerge in a political system possessing democratic institutions at cross-purposes. One line of thought, beginning with Dahl [22], suggests that the Supreme Court of the United States has a unique capacity among major institu-

tions of American government to leverage its legitimacy in order to change mass opinion regarding salient policies. If the Dahl's hypothesis is correct, then the Supreme Court's same-sex marriage decisions should have resulted in a measurable change in opinion. A primary finding about implication of Dahl's hypothesis is that the Court is polarizing, creating more supportive opinions of the policies it reviews among those who supported the policy before the decision and more negative opinions among those who opposed the policy prior to the decision [30] [47].

We propose more fine-grained dimensions for political sentiment analysis, such as supportiveness, emotional intensity and polarity, allowing political science researchers, policy makers, and the public to better comprehend the public reaction to major political issues of the day. As we describe below, these different dimensions of discourse on Twitter allows examination of the multiple ways in which discourse changes when the Supreme Court makes a decision on a given issue of public policy. Our dimensions also open the door to new avenues of theorizing about the nature of public discourse on policy debates.

We present a case study in which our results might be used to answer core questions in political science about the nature of Supreme Court influence on public opinion. Political scientists have long been concerned with whether and how Supreme Court decisions affect public opinion and discourse about political topics [41] [47] [33]. Survey research on the subject has been limited in two ways. Survey analysis, including panel designs, rely on estimates near but never on the date of particular decisions. In addition, all survey-based research relies on estimates derived from an instrument designed to elicit sentiment – survey responses, useful as they are, do not reflect well how public opinion is naturally expressed. Our analysis allows for the examination of public opinion as it is naturally expressed and in a way that is precisely connected to the timing of decisions.

Next, we state the problem more formally, and outline our approach and implemen-

tation.

5.1.1 Political Sentiment Classification

We propose three refinements to sentiment analysis to quantify political opinions. Specifically, we pose the following dimensions as particularly important for politics:

- Support: Whether a Tweet is ***Opposed***, ***Neutral***, or ***Supportive*** regarding the topic.
- Emotional Intensity: Whether a Tweet is emotionally ***Intense*** or ***Dispassionate***.
- Sentiment Polarity: Whether a Tweet's tone is ***Angry***, ***Neutral***, or ***Pleased***.

In this work, each of the proposed measures is treated as a supervised classification problem. We use multi-class classification algorithms to model Support and Sentiment Polarity, and binary classification for Emotional Intensity and Sarcasm. Section 5.2 describes the labels used to train the supervised classification models. Notice some classes are more interesting than the others. For example, the trends or ratio of opposed vs. supportive Microblogs are more informative than the factual ones. Particularly, we pay more attention to the classes of *opposed*, *supportive*, *intense*, *angry*, and *pleased*.

5.1.2 Classifier Feature Groups

To classify the Microblog message into the classes of interest, we develop 6 groups of features:

Popularity: Number of times the message has been posted or favored by users. As for a Tweet, this feature means number of Retweets and favorites.

Capitalization and Punctuation.

N-gram of text: Unigram, bigram, and trigram of the message text.

Sentiment score: The maximum, minimum, average and sum of sentiment score of terms and each Part-of-Speech tags in the message text.

Counter factuality and temporal compression dictionary: This feature counts the number of times such words appear in the message text.
Political dictionary: Number of times a political-related word appears in the message text.
We compute sentiment scores based on SentiWordNet[1], a sentiment dictionary constructed on WordNet.[2] Political dictionary is built upon political-related words in WordNet. As in this section, we construct a political dictionary with 56 words and phrases, such as "liberal", "conservative", and "freedom" etc.

The classifiers we tested include Naïve Bayes, Maximum Entropy, Support Vector Machine, and the decision tree. Classification results computed by cross-validation show that Naïve Bayes is surprisingly robust to sparse and noisy training data in our dataset (details of the dataset is in Section 5.2).

5.2 Case Study: Defense of Marriage Act

Our goal is to build and test classifiers that can distinguish political content between classes of interest. Particularly, we focus on classifying Tweets related to one of the most popular political topics, "Defence of Marriage Act" or DOMA, as the target. The techniques can be easily generalized to other political issues in Twitter.

Dataset

In order to obtain relevant Tweets, we use Twitter streaming API to track representative keywords which include "DOMA", "gay marriage", "Prop8", etc. We track all matched Tweets generated from June 16th to June 29th, immediately prior and subsequent to the DOMA decision, which results in more than 40 thousand Tweets per day on average.

[1] http://sentiwordnet.isti.cnr.it/
[2] http://wordnet.princeton.edu/

Human Judgments

With more than 0.5 million potential DOMA relevant Tweets collected, we randomly sampled 100 Tweets per day from June 16th to June 29th, and 1,400 Tweets were selected in total. Three research assistants were trained and they showed high agreement on assigning labels of relevance, support, emotional intensity, and sentiment polarity after training. Each Tweet in our samples was labeled by all three annotators. After the labeling, we first removed "irrelevant" Tweets (if the Tweet was assigned "irrelevant" label by at least one annotator), and then the tweets with no major agreement among annotators on any of the sentiment dimensions were removed. As a result, 1,151 tweets with what we consider to be reliable labels remained in our dataset (which we expect to share with the research community).

Annotator Agreement The Fleiss' Kappa agreement for each scale is reported in Table 5.1 and shows that labelers have an almost perfect agreement on relevance. Support, emotional intensity, and sentiment polarity, show either moderate or almost perfect agreement.

Measure	Fleiss' Kappa
Relevance	0.93
Support	0.84
Intensity	0.54
Polarity	0.49

Table 5.1: Agreement (Fleiss' Kappa) of Human Labels.

Classification Performance Results

We reproduce the same feature types as previous work and develop the political dictionary feature for this particular task. We experimented with a variety of automated classification algorithms, and for this preliminary experiment report the performance of

Value	Prec. (%)	Rec. (%)	Accuracy(%)
Supportive (48%)	73	74	68
Neutral (45%)	76	67	
Opposed (7%)	17	30	
Intense (31%)	56	60	73
Dispassionate (69%)	81	79	
Pleased (10%)	48	31	69
Neutral (79%)	84	78	
Angry (11%)	24	45	

Table 5.2: Performance of Classifiers on Each Class.

Naïve Bayes algorithm (simple, fast, and shown to be surprisingly robust to classification tasks with sparse and noisy training data). 10-fold cross validation are performed to test the generalizability of the classifiers. Table 5.2 reports the average precision, recall and accuracy for all measures. Sarcasm is challenging to detect in part due to the lack of positive instances. One goal in this study is to build a model that captures trends among the different classes. In Section 5.2, we will show that the trends of different measures estimated by the trained classifier align with the human annotated ones over time.

Visualizing Sentiment Before and After DOMA

One natural application of the automated political sentiment analysis proposed in this section is tracking public sentiment around landmark U.S. Supreme Court decisions. To provide a more reliable estimate, we apply our trained classifier on all relevant Tweets in our collection. More than 2.5 million Tweets are estimated in four proposed measures. Figure 5-1 shows the distribution of on-topic Tweet count over time. The Supreme Court decision triggered a huge wave of Tweets, and the volume went down quickly since then.

Figures 5-2 and 5-3 visualize both the human labeled trends and the ones obtained by the classifier for the classes “Supportive” and “Intense”. In both figures, the peaks in the predicted labels generally align with the human-judged ones. We can see the

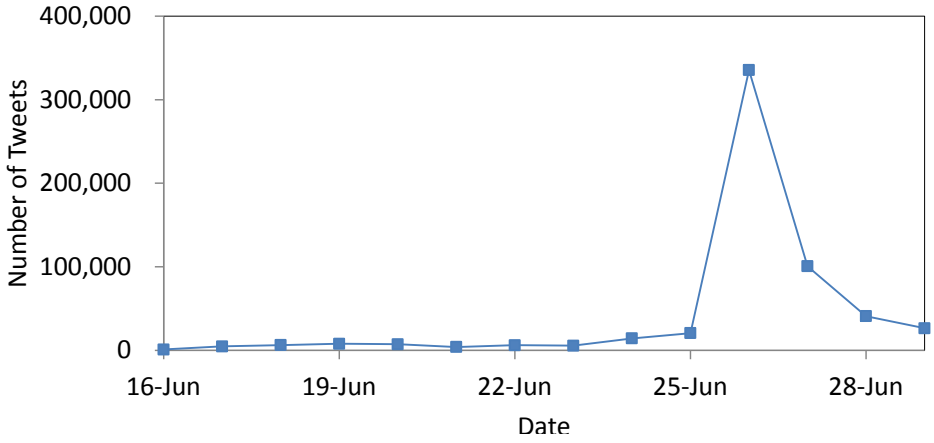

Figure 5-1: Number of "Gay Marriage" Tweets Over Time.

supportiveness and intensity are both relatively high before the decision, and then they decline gradually after the Supreme Court decision.

Figure 5-3 shows the volume of intensive Tweets detected by our trained model has a burst on June 22rd, which is not captured by human labeled data. To investigate this, we manually checked all Tweets estimated as "intensive" on June 22rd. It turns out most of the Tweets are indeed intensive. The reason of the burst is that one Tweet was heavily retweeted on that day. We do not disclose the actual tweet due to its offensive content.

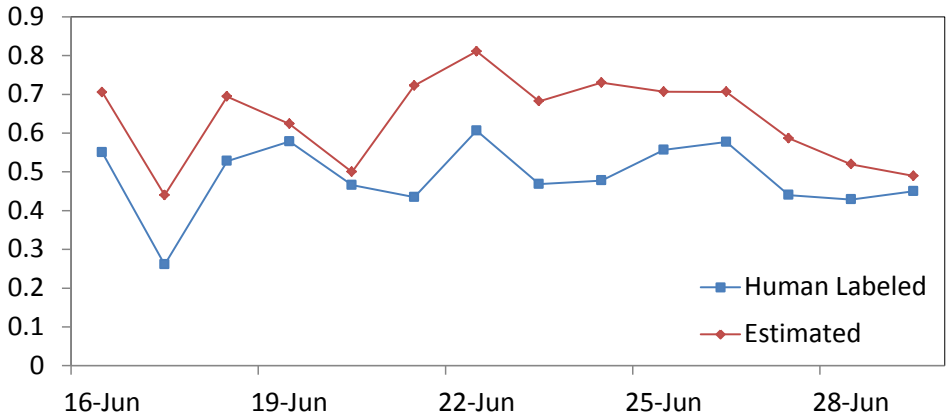

Figure 5-2: Percentage of "Supportive" Tweets Over Time.

Figure 5-4 plots the trends of "supportive" and "opposed" Tweets in different scales. According to the Supreme Court decision, the "supportive" group wins the debate. Inter-

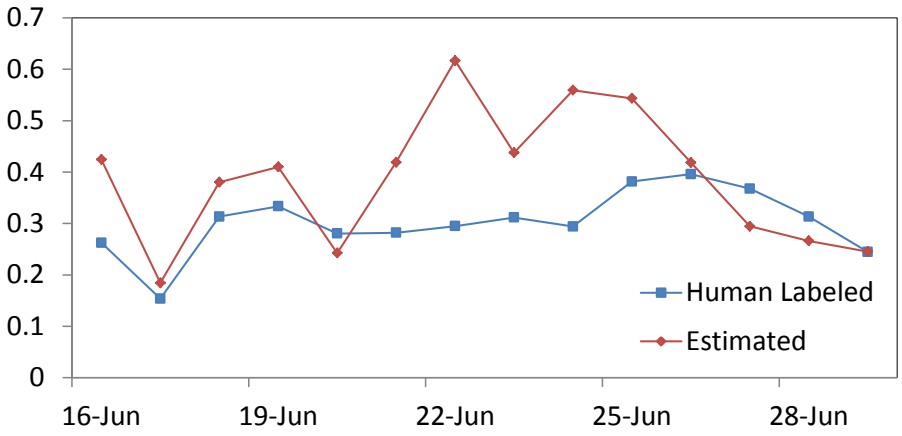

Figure 5-3: Percentage of "Intense" Tweets Over Time.

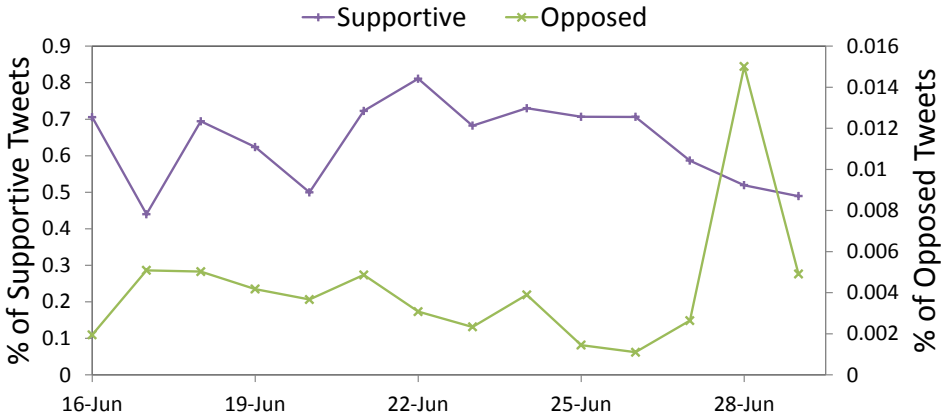

Figure 5-4: Comparison between "Supportive" and "Opposed" Trends.

estingly, instead of responding immediately, the "loser" group react and start Tweeting 2 days after the decision. These trends indicate that "winner" and "loser" in the debate react differently in time and intensity dimensions.

We believe that our estimates of sentiment can be used in various ways by political scientists. The "positivity bias" [32] model of Supreme Court opinion suggests that the Court can move public opinion in the direction of its decisions. Our results possibly indicate the opposite, the "polarizing" model suggested by [30] and [47], where more negative opinions are observed after the decision (in Figure 5-4), at least for a short period. By learning and visualize political sentiment, we could crystalize the nature of

the decision that influences the degree to which the Supreme Court can move opinion in the direction of its decisions.

5.3 Inferring Latent User Characteristics for Analyzing Sentiment

As demonstrated in previous section, social media has become an important resource for political science analysis. Platforms like Twitter carry the opinions of millions of users on a variety of political discussions and are available in real-time. While this has afforded enormous research opportunities, the nature of the data provide challenges as well. Demographic characteristics are a key factor in political science research, yet they are missing from Twitter. Previous work has compensated for this challenge by relying on supervised learning to construct classifiers for traditional demographic characteristics, such as age, gender and ethnicity. We believe this approach presents challenges for practical political science research: classifiers require manually labeled training data and are often domain specific. Additionally, it misses a significant opportunity: the ability to model finer grained demographic attributes than what is normally available from survey data.

In this section, we use unsupervised learning to discover user characteristics directly from Twitter data. Our methods rely on user self-descriptions: short snippets of user authored text available in their profile. Additionally, we consider how profiles of contacts in the social network can further inform the characteristics of a user. We demonstrate the efficacy of our approach by analyzing two major political issues, both of which became major topics of discussion following US Supreme Court decisions in June of 2013. We consider three major analyses: an intrinsic evaluation of the learned characteristics, an extrinsic evaluation of the characteristics in an analysis of the two political topics, and an analysis of the new demographic groups that arise from these characteristics. In all cases, our approach improves over using traditional demographic attributes. This suggests a promising new research direction in support of political and social science analysis of social media discourse.

5.3.1 Motivation

The proliferation of social media websites, such as Twitter, offers researchers new opportunities to understand public opinion towards various issues. Analyzing user comments and opinions concerning political topics can aid political science research and government policy making. Estimating user sentiment and identifying opinion groups provides a means to quantitatively measure public opinion.

While raw sentiment and opinions are important, they are only part of the story. For information to be meaningful it must be contextualized within a socio-economic and cultural frame. We care not just what the opinion is, but who has it. The use of demographic categories (e.g., gender, age, ethnicity) to characterize and group users, has been well studied and established in social science. Furthermore, public opinion scholarship has consistently found that demographics are strong predictors of political values [28, 52]. However, accurate demographic information is often unavailable on social media platforms such as Twitter.

The predictive value of cultural groups for opinions, and their importance in analyzing discovered opinions, offers an opportunity. By predicting these groups we can aid in the discovery and identification of sentiment and opinions. Consider the example in Figure 5-5, which shows two similar comments written by users with contrasting political views about the same U.S. Supreme Court decision. On their face, these two comments express the same opinion about a court decision; however, the conflicting user profiles suggest that one of them is sarcastic. While many models have been proposed to estimate users' political sentiment from social media [76], relying on these short and ambiguous texts alone make this problem challenging. However, when the user's demographics or characteristics are considered the task becomes clearer. Understanding the various characteristics of users add necessary context to both the automated sentiment task, and the analysis conducted by political science researchers. When users share similar demographics, attributes, beliefs, political leanings, occupations, and interests, they

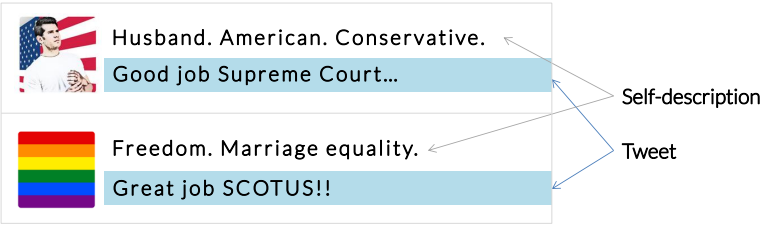

Figure 5-5: Synthetic users' comments on U.S. Supreme Court decision of Defence of Marriage Act.

often share opinions towards political issues. The great diversity of these characteristics suggest that, rather than relying on predetermined demographic categories, these characteristics should be inferred directly from data. Additionally, reliance directly on raw data relies the burden of creating supervised resources for demographic attribute classification.

In this section, we demonstrate that user characteristics identified directly from social media data provide better context for the automated analysis of political discussions. We consider characteristics derived from user self-descriptions, a resource heretofore unexplored, although other work has relied on self-descriptions in the posts themselves [9]). We rely on unsupervised learning to infer a variety of characteristics for each user based on this information.

The use of self-descriptions could bring practical benefits: (1) the content of self-descriptions are usually more stable than users' posts, which can help build more robust models of sentiment estimation; (2) the resulting characteristics represent users' attribute groups and therefore can be easily interpreted; (3) self-descriptions can cover a wider range of information than established demographic categories, which may help better characterize Twitter users. However, self-descriptions on Twitter can be short and brief, and sometimes inadequate to comprehensively represent users' characteristics. To overcome this challenge, we additionally consider self-descriptions from the social network. By leveraging the follower connections on Twitter, characteristics for users can

still be robustly inferred even if the user lacks a self-description.

We demonstrate the efficacy of using inferred user characteristics for political analysis of social media as follows. First, we describe our method for inferring user characteristics baed on self-descriptions of the user and the social network. We then conduct an intrinsic analysis to demonstrate that these methods lead to human interpretable characteristics, some of which capture conventional demographic categories. In an extrinsic analysis using these characteristics for a political discussion analysis task, we show how using these characteristics improves sentiment accuracy, even compared to the inclusion of features based on supervised demographic attribute classification. Finally, the analysis of the resulting opinion groups suggests that our user characteristics produce more coherent opinion groups than human-annotated demographic attributes.

5.3.2 Methodology

We consider how a user can best be represented for an analysis task. Roughly speaking, there are two ways two represent a user: based on their authored content (tweets) and based on the demographic information of the user. The first methods is a popular choice, and makes sense for tasks that rely on the topics of posts. However, the importance of demographic attributes in political analysis suggests that the second may provide additional valuable information. While previous work has relied on supervised classifiers for obtaining traditional demographic characteristics [12, 16, 21, 58, 62, 67, 70, 90], we consider a method for automatically inferring user characteristics directly from data.

Learning Algorithm

Our goal is the unsupervised discovery of user characteristics based on available data for each user. We make the following modeling assumptions. Each user can be represented by one or more latent characteristics z. Furthermore, these characteristics may not equally describe the user. Instead, we assume that each user d has a distribution of

characteristics: θ_d. Overall in the corpus, some characteristics may be more popular, such as gender characteristics. We capture this by having each user's distribution over characteristics θ_d drawn from a Dirichlet distribution parameterized by the vector $\bar{\alpha}$.

As we will describe below, each user is represented by an observed collection of words $\mathbf{w}_d$. Based on these words, we need to infer the distribution θ_d. In our model, each word $w \in \mathbf{w}_d$ is generated by first selecting a characteristic z and then selecting a word w from ϕ_z, a distribution over all words specific to characteristic z. In this model, we learn to associate certain words with each characteristic (e.g. words that indicate gender) and encode these as a distribution. In summary, we can generate the observed words for each user by first selecting a distribution over characteristics for the user θ_d. Then for each word to generate, we select a characteristic $z \sim \theta_d$, and $w \sim \phi_z$.

Note that this model assumes that a user has multiple characteristics, not a single defining one. It is therefore an ad-mixture model, as opposed to a mixture (clustering) model. Previous work focused on user clustering, which assigned a single cluster or label to each user [29, 54]. While one could create clusters of users based on their shared characteristics, the characteristics model more accurately reflects actual users.

Our model of user characteristics is equivalent to latent Dirichlet allocation (LDA) [15], where users are documents and characteristics are topics (note that we follow standard LDA notation for clarity.) Therefore, we learn user characteristics with a standard LDA implementation.)

User Descriptions

We mine descriptions about each user provided by the user's themselves. Twitter users can optionally provide a profile, which contains a free text description of the user. The variety of content in these profiles include a range of characteristics of interest: occupation, religion, political leanings and gender. Other users fill the description with less informative information, such as mottos, quotes, song lyrics or jokes. These may also

be informative of the user, but they are less so.

Figure 5-5 gives two illustrative user descriptions: “Husband. American. Conservative.” and “Freedom. Marriage equality.” Despite being very short, these descriptions are incredibly informative. The first can directly inform gender and political affiliations and the second provides strong clues to political affiliation. While short texts present significant challenges to topic models, the density of information may yield more informative characteristics.

We utilize user descriptions by representing each user d as a short document containing only their description. For users without profiles we assume a uniform distribution θ_d.

Social Network

While user descriptions can be highly informative, they are not always so. As noted above, many users write uninformative descriptions. Still others exclude them entirely. Even when they are informative, they are still quite short, which poses challenges for algorithms that rely on word co-occurrence statistics, such as LDA.

A more robust source of self-descriptions is the social network surrounding a user. Others have found that social network users tend to be connected with similar users, where similarity can reflect shared attributes or beliefs [11]. The observed “homophily” of the Twitter social network [45] means that characteristics of a user can be inferred based on other connected users. Utilizing descriptions from the social network in place of the user’s available data allows us to over come data sparsity in the cases described above.

There are numerous ways to measure social connections on Twitter, such as identifying “@” mentions and follower lists. We rely on the list of all users that are followed by the given user. For convenience, we term this set the *friends* of the user.

We infer characteristics using friends as follows. We learn an LDA model on user

self-descriptions as described in the previous section. This provides a distribution θ_d for every user in the collection. For each user, we collect all their friends as represented by the corresponding distributions $\theta_{d'}$ for all $d' \in \text{friends}(d)$. We create a new distribution $\hat{\theta}_d$ for each user as the average of their friends distributions: [3]

$$\hat{\theta}_d = \frac{1}{|\text{friends}(d)|} \sum_{d' \in \text{friends}(d)} \theta_{d'} \tag{5.1}$$

An alternative way to represent a user's profile is to aggregate all her friends' self-descriptions into a single document. However, this approach has several drawbacks: (1) assembling an aggregated description would lose original document-level information, where each aggregated profile represent a group of people instead of one; (2) this approach would weigh friends by the length of their descriptions. In contrast, our approach weighs friends equally by inferring friends' characteristics first, and then aggregating the resulting distributions.

The result of this method is that we can represent a user with their distribution θ_d learned from their self-descriptions and the self-descriptions of their friends.

5.3.3 Evaluation Setup

Our goal is to demonstrate that the inferred user characteristics aid in the research of political issues on Twitter. We consider two major political events in the United States in 2013 that centered around controversial Supreme Court decisions. Both opinions were released in June 2013.

- Same-sex marriage (SSM): The court ruled that the Defense of Marriage Act (DOMA) was unconstitutional. (Keywords: "same-sex marriage", "DOMA", "gay marriage".)

[3] We experimented with adding the user's own description as well, but excluding it performed better. We include some of these results in the experiments.

- Voting rights act (VRA): The court struck down Section 5 in VRA as unconstitutional. (Keywords: “VRA”, “voting rights act”.)

Data Collection

Before we evaluate our inferred characteristics, we describe our data collection. To analyze the discussions around the two political topics above we obtained data using the Twitter API in two ways.

First, we collected a random sample of Twitter users that represent the general Twitter population, regardless of participation in political discussions. We obtain tweets for the first 21 days of April 2013 via the Twitter streaming API, which provides a random 1% sample of public tweets. We divide this data into three sequential batches, each with 7 days. Each batch yielded roughly 36 million tweets and 15 million unique usernames. From these usernames, we sampled 2 million usernames per batch (6 million total) to constitute our collection of random Twitter users. During sampling, we only select users who have a self-description of more than 3 words after removing English stopwords. We chose to create three samples from different weeks for a more robust evaluation. Since different topics trend at different times, user participation changes as well. By modeling users from three different time periods, we can evaluate robustness of our model to possible changes in user makeup. We refer to this collection as our background sample.

Second, we collected data specifically around the political topics described in the previous section. We used the associated keywords to collect tweets via the Twitter streaming API for all of 2013. The resulting collection contained hundreds of thousands of unique users. As part of our evaluation we labeled users for sentiment and demographic attributes. Users to annotate were randomly drawn by sampling from a two week period for each of the political topics based on their associated keywords. A summary of the tweets per political topic and number of annotated users is shown in

		SSM	VRA
Total	Tweets	2,473,482	37,074
	Users	689,570	17,241
	Friends	2,000,000	2,000,000
Annotated	Users	854	702

Table 5.3: Number of tweets and users collected and annotated for the Same-Sex Marriage (SSM) and Voting Rights Act (VRA) topics.

Table 5.3. We refer to this collection as our political sample.

To infer characteristics based on the social network, we need to obtain data for friends of users. Collecting all friends for all users would be too time consuming with Twitter API rate limits. We collected friends information in two ways. First, to ensure we had the full friends information for the annotated users, we collected all of their friends profiles. Second, we collected friends for a random sample of 10,000 users for each of the SSM and VRA datasets. To cover all the friends of these users, we would have to download a total of 4 million user profiles for each dataset. Instead, we randomly downloaded half that amount: 2 million for each dataset. In our experiments, we will use the first batch (friends of annotated users) for our evaluations, and we will use the second batch (2 million friends of 10,000 sampled users) for training our models.

User Annotation

Our analysis will consider both the sentiment towards the political topic of a user, and the demographic characteristics of the user. To support this evaluation we obtained human labels for both of these tasks on the users sampled for annotation described above.

Sentiment We labeled users with regards to their sentiment towards the political topic. Users were classified as either being supportive of, neutral towards, or in opposition to the political topic, e.g., opposed to same-sex marriage. Initially, we attempted to use Amazon Mechanical Turk to obtain labels for this task. However, correct an-

notation required political background knowledge of these issues, which we found the turkers did not have. Instead, we relied on three annotators recruited directly by the authors to label the data. Annotators were shown the user profile information (i.e., name, self-description, profile image), the tweets from the political dataset relevant to the topic, and were given a link to the user's Twitter page.

Demographics Tweets were labeled with regards to four demographic categories:

1. **Gender**: male, female
2. **Age**: teenage, young adult, adult, middle age
3. **Ethnicity**: caucasian, african american, hispanic, asian
4. **Education**: high school or below, some college, college graduate

Annotations were obtained using Amazon Mechanical Turk. For each user, we displayed the same information as shown for the sentiment task. Annotators were instructed to indicate if not enough information was provided to make a determination for a demographic attribute. We obtained three sets of annotations for each user. We took the gold label as the one on which two of the annotators agreed.

We observed high agreement on both political datasets for sentiment. User attributes were more challenging since users often do not provide clear indicators of these attributes in their profile or tweets. To further evaluate agreement on those users who may have provided this information, we first filter out demographic attributes on which the three annotators completely disagreed, i.e., we kept demographic attribute labels for which two annotators agreed. We measured agreement between annotators using Fleiss' Kappa on the attribute labels that remained after filtering (Table 5.4).

Gender, age, and ethnicity have reasonable agreement, likely because this information often appears in the profile. However, since education is not usually attested to in the same way, it proved difficult to label. Because of the low agreement we did not

		Fleiss' Kappa		% remaining	
		SSM	VRA	SSM	VRA
Sentiment		0.84	0.89	-	-
Attributes	Gender	0.68	0.63	0.69	0.55
	Age	0.33	0.37	0.52	0.38
	Ethnicity	0.43	0.68	0.59	0.42
	Education	0.25	0.13	-	-

Table 5.4: Annotators' agreement (Fleiss' Kappa) for the annotation tasks, as well as the percentage of tweets that remained after filtering for majority agreement on the attributes task.

consider it in our evaluations. For the remaining three demographic categories, we used only those instances that had a majority agreement, where at least two annotators agreed on the label (rightmost 2 columns of Table 5.4.) This percentage of tweets that remained after filtering is another measure of annotation difficulty.

For our evaluation, we only used users who had agreement on all three of the demographic attributes utilized in our experiments: gender, age and ethnicity. In total, we had 367 users for SSM and 274 users for VRA.

Inferring User Characteristics

For the experiments described below, we obtain user characteristics as follows. We use the LDA implementation provided in JGibbLDA [69]. For training we used 1000 iterations with $\bar{\alpha} = 0.1$ and $\bar{\beta} = 0.1$. We removed English stop words from all texts.

We begin by inferring characteristics on the first batch of background users: 2 million users. We only used users that have at least three words in their self-description after stop word removal. We repeat model learning for both the second and third batches as well. We used 500 topics, which we found to be a sufficient number after early experimentation with different numbers of topics. In general, we favor fine-grained characteristics (more topics) over a smaller number of coarse topics. To infer characteristics for users in the political sample we use the model trained on the first batch, and then

estimate characteristics for the political samples without updating the model parameters. Experiments that use characteristics inferred from a different procedure will be indicated below.

Some of our experiments will use the content of the tweets as a baseline. We process the tweet content so that for each user, we collect all available posts by that user into a single document for LDA.

5.3.4 Intrinsic Analysis

We demonstrate the impact of our inferred characteristics on the two political topics described above in three major parts. First, we conduct an intrinsic evaluation of the inferred characteristics, demonstrating that they correspond to interpretable attributes. This includes measuring their stability over different datasets. Second, we conduct an extrinsic evaluation by showing how the characteristics can be used to analyze political discussions. This includes measuring the impact of attributes on the participation rate in political discussions, and their effectiveness in predicting sentiment as compared to standard demographic attributes. Third, we analyze the group homogeneity resulting from our inferred characteristics to show that they identify informative groupings.

Robustness of User Characteristics

We begin our intrinsic analysis of the inferred characteristics by showing that characteristics are robust across multiple samples of users.

Figure 5-6 shows the top 10 most popular characteristics in the first background sample inferred from user self-descriptions. Popularity is measured by assigning each user to the characteristic with the highest probability in the user's θ_d. The figure shows the three most likely words for each characteristic according to ϕ_z as well as a label we assigned based on a manual review of ϕ_z. The most popular characteristic covers about 70,000 of the 2 million users, whereas the least popular one (not shown) covered

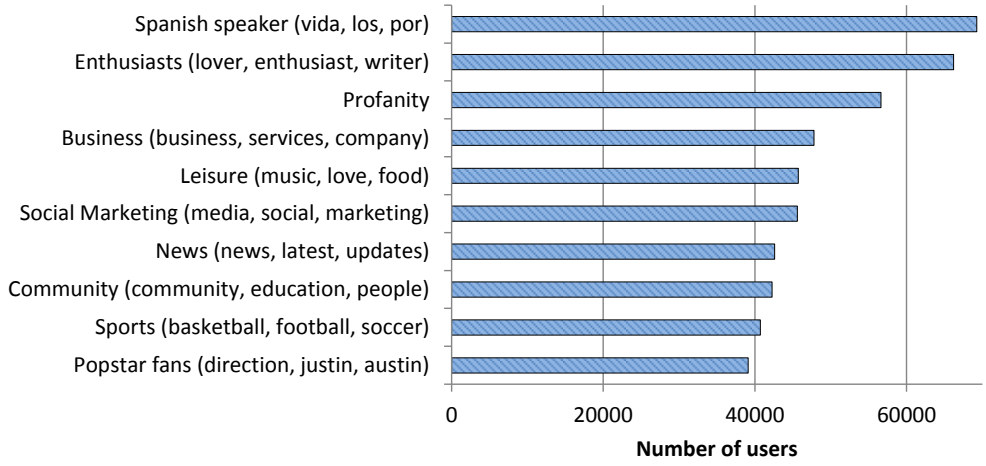

Figure 5-6: Top 10 user characteristics inferred from random sampled users. Vertical axis is formatted as: Characteristic name (3 top words).

2,500 users. We found that the popular characteristics were semantically coherent and therefore easy to interpret.

We seek characteristics that can be used for a variety of different political analyses. Therefore, we ask: how sensitive are the inferred characteristics to the training data sample? We measure this sensitivity by comparing the characteristics learned between the first batch of background data compared with the second and third batches for user self-descriptions. We match characteristics across the batches by measuring the L_1 distance between the distributions ϕ_z for all pairs of characteristics. We use L_1 distance instead of other metric (e.g., KL-divergence) because L_1 gives a bounded range of similarities (i.e., from 0 to 1), which can help identify "perfect alignment" and "zero alignment". We greedily align each characteristic from batch 1 to a single characteristic from each of batches 2 and 3. We visualize the resulting alignments by grouping the characteristics into bins of size 50 ordered by the number of users with each characteristic in the first batch, e.g. the first bin contains the 50 most popular characteristics in the first batch. For each bin, we compute the average L_1 distance for all of its pairs. Figure 5-7 shows these ten bins and their average L_1 distance for the comparison between the first and second batch, and the first and third batch. The popular characteristics, those

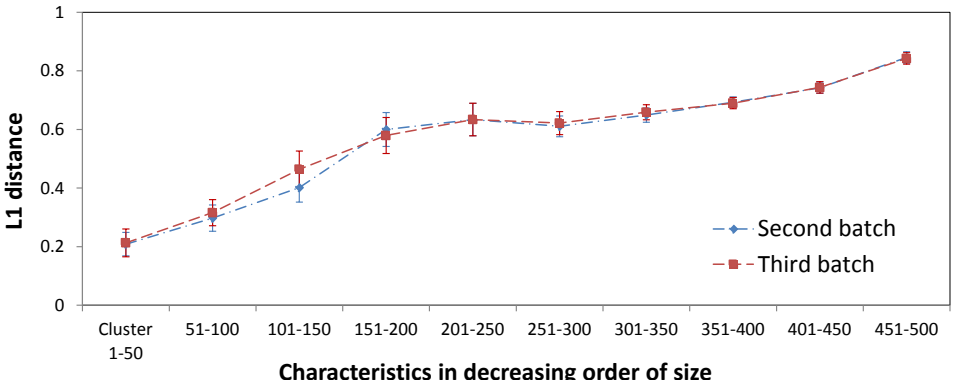

Figure 5-7: L_1 distance between clusters inferred from original sampled users and two additional batches of resampled users.

that we most want to be consistent across samples, have the lowest L_1 distance, which suggests that these 50 characteristics are well represented in each batch. As we consider less popular characteristics, we find worse matches across batches. A manual analysis of the pairs found that those with an L_1 distance below 0.45 were well matched and had consistent interpretations. Averaged across both batch alignments, 153 characteristics had a pair score below 0.45 and these covered about 50% of the 2 million users in each batch. These results indicate that there are sufficient robust characteristics that can extend across different datasets.

Interpretability of User Characteristics

We now conduct a qualitative analysis of the inferred characteristics. We presented an annotator with the top 10 words for each of the 500 characteristics inferred on the first batch using user self-descriptions. We asked the annotator to name each of the characteristics with a descriptive label. The annotator was able to label 305 of the 500 characteristics with a label; the remaining characteristics did not appear coherent. We found that most of the incoherent characteristics originated from self-descriptions that contained song lyrics and quotes. While they appear incoherent, these characteristics do link together many users who may have shared interests; they may be useful as fea-

tures in an automated evaluation. We manually assign these characteristics into broader categories to find out what type of information they can provide. Using the label provided by the annotator, we grouped the characteristics into broad categories. Examples of these categories, some labeled characteristics and their top words appear in Table 5.5.

Many of the characteristics correspond directly to demographic attributes, e.g., gender, age, education, etc. Demographics have consistently been found to be important predictors, confounders or moderators of political values in political science [28, 52]. However, researchers believe that demographics themselves do not induce particular opinions, rather contextual factors, such as people's life experiences, which are shaped by demographics are most formative in shaping political opinions [80]. We note that the model identifies characteristics that do not fit into traditional demographic groups, such as life experience. The ability to learn these non-traditional characteristics, as well as many fine-grained ones, can impact political science analyses.

5.3.5 Extrinsic Analysis

We now focus on an extrinsic evaluation of the user characteristics: can they improve tasks associated with an analysis of political discussions on Twitter? Since we include several quantitative tasks we can directly compare the efficacy of learning characteristics from self-descriptions with traditional demographics.

Measuring Discussion Engagement

The level of interest in a political topic can be gauged by measuring the level of engagement in an associated discussion. A common question is to evaluate the level of interest of different populations in political issues. Since demographic information is correlated with interest in certain political issues, our user characteristics should be informative in determining discussion engagement.

Using the same metric for determining characteristics popularity as above (section

Category	Label	Top words
Gender	Female	wife, mom, mother
	Male	husband, father, dad
Age	Teenager	girl, guy, teenage
	Veteran	retired, veteran, vet
Education	High school	high, school, class
	College	student, college, major
	Researcher	university, professor, research
Occupation	Entrepreneur	founder, CEO, entrepreneur
	Professionals	coach, consultant, expert
	Lawyer	law, legal, lawyer
	Developer	web, developer, software
Political affiliation	Conservative	conservative, Christian, #tcot
	Liberal	liberal, progressive, political
Emotions	Positive	peace, rest, happiness
	Negative	fat, depression, anxiety
Life experience	Past	found, loved, lost
	Relationships	amazing, boyfriend, girlfriend
	Life wisdom	lives, change, inspire
Nationality	Mexican	vida, soy, por
	Dutch	van, met, voor
	Swedish	och, med, som
	French	les, des, pour

Table 5.5: Identified user characteristics and their associated categories.

5.3.4) and the characteristics inferred for the users in the political sample (section 5.3.3), we computed the percentage of users with each characteristic in each of the political datasets. As a baseline, we use the percentage of users with each characteristic in the first batch of the background sample.

Figure 5-8 shows the relative difference between each political sample and the background sample for the top ten most common characteristics (section 5.3.4). Several of these characteristics stand out as very over or under represented in the political discussions. The characteristics of *community*, *news*, and *social marketing* show up much more frequently in the voting rights act discussion, though for the same-sex-marriage discussion they do not substantially differ from the background. *Spanish speakers*, *sports*, and *popstar fans* are less interested in these two political discussions. Moni-

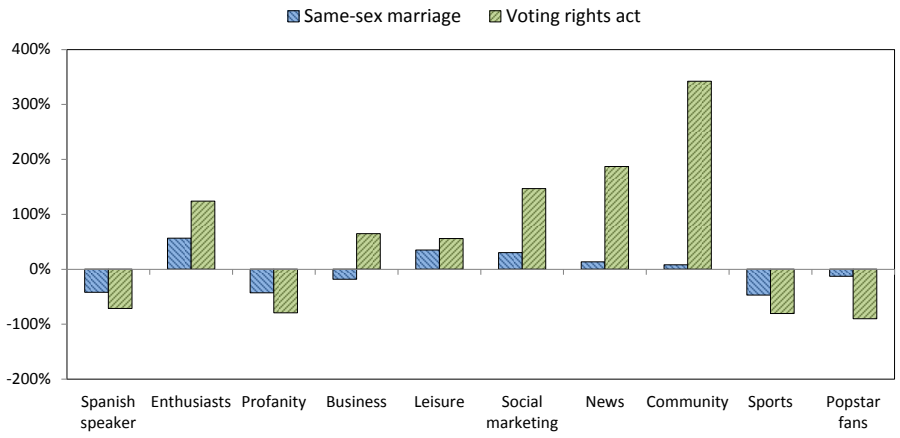

Figure 5-8: Relative difference (compared to background users) of discussion participation rates in the ten most common characteristics.

toring the involvement of characteristics over time could reveal changes in engagement, distinguishing short-term participation versus persistent involvement.

Estimating Political Support

Beyond engagement, we wish to know what opinions on political issues are held by different groups. In this work, we consider this as the task of determining if a user is supportive of, neutral towards, or in opposition to a political position (sentiment). We model this as a classification problem and construct multi-class classifiers that predict these three labels based on a variety of features. In our experiments, we use SVMs with polynomial kernels, which consistently provided the best results when evaluated against several other types of classifiers. We used the Weka machine learning toolkit [40] and relied on the default parameters of all experiments. We evaluate using 10-fold cross validation on the 367 SSM and 274 VRA users. For each experiment we report the accuracy and area under the ROC curve, as well as the majority baseline.

We begin with a comparison of different methods for representing user information only, not the contents of their tweets. Obviously, knowing what a user says can assist in predicting their opinion; our next experiments will consider these baselines. First, we

limit our evaluation to considering what method best represents a user for the political sentiment task.

We compare the effectiveness of using inferred user characteristics for representing a user against several baseline approaches.

- **Demographics**: We represent a tweet's author with features indicating their gender, age, and ethnicity. We note that we use the annotations provided by the turkers. This is an overly optimistic baseline as a realistic evaluation would rely on automated methods for determining these demographics. Nevertheless, we chose to use the human provided labels to determine an upper bound on the performance of a demographics system.

- **N-grams**: Our model uses self-descriptions to infer characteristics. We alternatively consider representing the self-descriptions as unigram and bigram features. Additionally, we consider n-grams based on all of the friends of a user.

We consider two information sources for inferring characteristics for users: the self-descriptions of each individual user, and the self-descriptions of the friends of each user (Section 5.3.2). We use each dimension of the distribution over characteristics as a real-valued feature.

Additionally, we consider different training sets for LDA. Previously, we trained LDA only on the background sample. However, including the users in the political dataset of interest can help to improve the inferred characteristics since they will be specific to the analysis task. On the other hand, only considering data from the analysis task may limit the amount of data available for learning.

We alter the training corpus for LDA in several ways.

- **Participants**: Only those users who are included in the relevant political dataset. This includes all users in the dataset (Table 5.3).

		Same-sex marriage		Voting rights act	
		Accuracy	AUC	Accuracy	AUC
Majority baseline		0.554	-	0.566	-
Demographics (gender, age, ethnicity)		0.630	0.649	0.546	0.488
Ngrams	User	0.532	0.554	0.570	0.550
	Friends	0.610	0.603	0.625	0.616
Characteristics	Participants	0.555	0.606	0.584	0.575
	Participants+Friends	0.537	0.586	0.599	0.599
	Background	0.515	0.556	0.584	0.579
	Participants+Background	0.537	0.578	0.562	0.542
	Friends	**0.640**	**0.700**	**0.628**	**0.652**
Tweet content	Ngrams	0.603	0.639	0.576	0.582
	LDA	0.601	0.636	0.584	0.602
Content +	LDA+Demographics	0.603	0.643	0.54	0.558
Characteristics	LDA+Friends	**0.657**	**0.712**	**0.602**	**0.620**

Table 5.6: Political sentiment classifiers results based on 10-fold cross-validation.

- **Participants+Friends**: Participants and the additional available friends of participants.
- **Background**: Use users from the first background batch only and apply the trained model to the users. This is the method used in previous sections.
- **Background+Participants**: Both the participants and the background users.

For training "Participants" and "Participants+Friends" we used 200 characteristics instead of the 500 used for our other experiments. We reduced the number of characteristics since the datasets in these cases were much smaller than the background sample. Preliminary experiments with different numbers of characteristics did not produce a significant impact on the results.

Table 5.6 reports results for all evaluation settings. The majority label for SSM is supportive, while for VRA its neutral. The demographics baseline, which uses traditional demographic categories provided by human annotators, does reasonably well compared to the majority baseline for SSM, but not for VRA. Knowing the demographics of a user can provide a strong signal as to their opinion in some cases, but the

traditional demographics are ineffective in others. Next, n-gram representations of the self-descriptions both achieve improvements over the majority baseline when utilizing friends' characteristics, though not when based on the user's characteristics, suggesting that by aggregating multiple descriptions we can overcome data sparsity.

While the characteristics we infer have mixed results compared the the demographics and majority baseline, the Friends characteristics provide the best results in every case. Two factors may explain this result: adding more data to represent a user can overcome data sparsity, and that users with uninformative descriptions may have friends with more informative descriptions. Additionally, these improvements over the demographics features are especially encouraging since the demographics are based on human provided labels (i.e., not predicted labels) whereas our characteristics are learned using an unsupervised method.

Following up on these results, we conducted another experiments that combines a user's characteristics with their friends. However, this decreased performance as compared to using the friends characteristics. Nevertheless, there may be additional ways to integrate both source of information, such as new models that integrate social network information directly into the inference process. Our primarily goal is to demonstrate the effectiveness of the information, rather than introduce new models. We leave this for future work.

Finally, we consider how user characteristics can improve sentiment classification when combined with the content of the user's tweets. We consider two methods of representing tweet content.

- **Ngrams**: We extract unigram and bigram features from all tweets by a user that are included in the political dataset.

- **LDA**: We train an LDA model on all tweets in the political dataset (one model per dataset.) We used the same learning setup as before with 200 topics. As described

Characteristic	Top words
Same-sex marriage	
Dream chaser	dream, big, true
Religious	God, Jesus, Christ
Profanity	
Teenager	girl, boy, teenage
Business	business, services, management
Voting rights act	
Conservative	dream, big, true
Religious	God, Jesus, Christ
Male	husband, father, dad
Veteran	retired, veteran, army
Female	married, kids, beautiful

Table 5.7: All of the most informative features were user characteristics, shown here in descending order according to information gain.

in Section 5.3.3, all tweets for a user are combined into a single document. We use the same number of characteristics as the best characteristic model (200). Tweets are then represented by the resulting topic distribution.

Table 5.6 compares these two approaches using the same setup as before. We find that LDA is a more effective representation of the tweet contents for this task. This is likely because of feature sparsity when relying on social media text.

Using the LDA representation of the content, we demonstrate that user characteristics can provide additional information helpful for this task. We consider two settings: adding traditional demographic information as features to the LDA based content features, and adding our best performing user characteristics (Friends) as features to the LDA based content features. The results (Table 5.6) show that while adding demographic features does not help, adding our characteristics features gives a significant improvement, yielding the best results overall for SSM, and while not improving over using only characteristics for VRA, it improves over using LDA and LDA+Demographics. The user characteristics encode valuable knowledge about the users that aids in determining the message's sentiment.

To determine which features are the most informative for learning, we ranked all the features used by LDA+Friends, the best performing model, according to their information gain for the label for each political dataset.[4] In both cases, the five most helpful features are user characteristics, shown in Table 5.7 with a manually assigned label and their three most likely words. Some of the characteristics represent demographic attributes, such as gender and age, while others capture more general attributes of the user (dream chaser) that are difficult to capture using traditional demographics. Additionally, the attributes correspond with common knowledge about these issues, e.g. religious individuals and young people are very likely to be opposed to and in favor of same-sex marriage respectively.

5.3.6 Group Homogeneity

Our final analysis will be to compare inferred user characteristics and demographic attributes in their ability to form homogeneous groups of opinions.

Group Opinion Homogeneity

Demographic analyses of political data divide users into demographic sub-groups, which often have a more predictable opinion towards the political topic than the overall population. For example, while the overall population may be evenly split on a topic, african-american middle-aged women may mostly agree on a single opinion.

Following this type of analysis, we determine if our user characteristics provide groups that have higher rates of agreement on political opinions than do standard demographic groups. Using the annotated sentiment data for the political datasets we measure opinion homogeneity as agreement around binary labels: supportive or opposing. We exclude users who were marked as neutral since they may have an opinion but it is not

[4]We note that since we are using a polynomial kernel, the feature space considered by the classifier will be a product of these features. Nevertheless, evaluating the individual features provides insight into which are informative.

shared in our data. We then compute the entropy over this binary label, where a lower entropy indicates a more homogenous group.

We construct groups in three ways.

1. Demographic groups for all possible combinations of the demographic attributes, i.e., a group is associated with an age, gender and ethnicity.

2. Using the Participants characteristics from Section 5.3.5, we assign each user to the most likely characteristic according to their distribution θ_d.

3. We assign groups as in (2) except we use the Friends characteristics. This is the best performing method in Table 5.6.

Rather than using all possible groups, we exclude those with only one user since the entropy of those groups would be zero.

Table 5.8 reports the number of groups and the average within-group entropy over the binary sentiment label. In all cases, our Friends based user characteristics improve over the demographic groups, yielding groups with more homogeneous opinions. Notice that higher number of groups cannot lower the entropy of groups with 2 or more users. A 2-user group with different opinions would simply achieve the highest possible entropy 0.693. Empirically, our results also suggest that higher number of groups does not lead to lower entropy, as the Participants method has the most groups but does not have the lowest entropy.

Interpreting User Characteristic Groups

Following up on our analysis that shows groups based on inferred user characteristics have more consistent opinions, we consider how they can be used in a demographic analysis of a political discussion. In this analysis we use the Friends characteristics. Figure 5-9 shows the support for each sentiment label in each demographic group in the

Grouping method	Groups with >= 2 users		Groups with >= 3 users	
	Number of groups	Average entropy	Number of groups	Average entropy
Same-sex marriage				
Friends	59	**0.163**	49	**0.168**
Participants	141	0.204	102	0.228
Demographics	21	0.257	17	0.277
Voting rights act				
Friends	53	**0.107**	35	**0.083**
Participants	91	0.134	59	0.137
Demographics	15	0.206	13	0.238

Table 5.8: Entropy of binary sentiment labels for each group as a measure of group homogeneity.

same-sex marriage discussion. There are clear differences between the groups: women are more supportive than men, young people are more supportive than older people, african american have the strongest opinions across all categories (highest support and opposition) and higher more educated individuals are less supportive.

Next, we rank the demographic groups from most supportive to most oppositional by measuring the number of users with each opinion in the group. Ties between groups are broken based on within group proportion of that opinion. Notice that we did not rank groups by percentage of opinion because many groups have very few users which makes them very easy to achieve 100% opinion percentage. However, these less popular groups are not representative. Thresholding the group size may help identify representative groups. But we need different thresholds for different political issues due to the distinct user distributions over groups. Instead of applying multiple complicated criteria, we find that groups ranked by number of users with each opinion simply provides reasonable results. Figure 5-10 shows the top supportive and oppositional characteristics for both political issues. For voting rights act, there is only one group that shows a clear opposition; it covers about 70% of all opposing users. The characteristics of progressive and politically active individuals were most supportive of more liberal policies in the context of same-sex marriage and voting rights, whereas the religious and con-

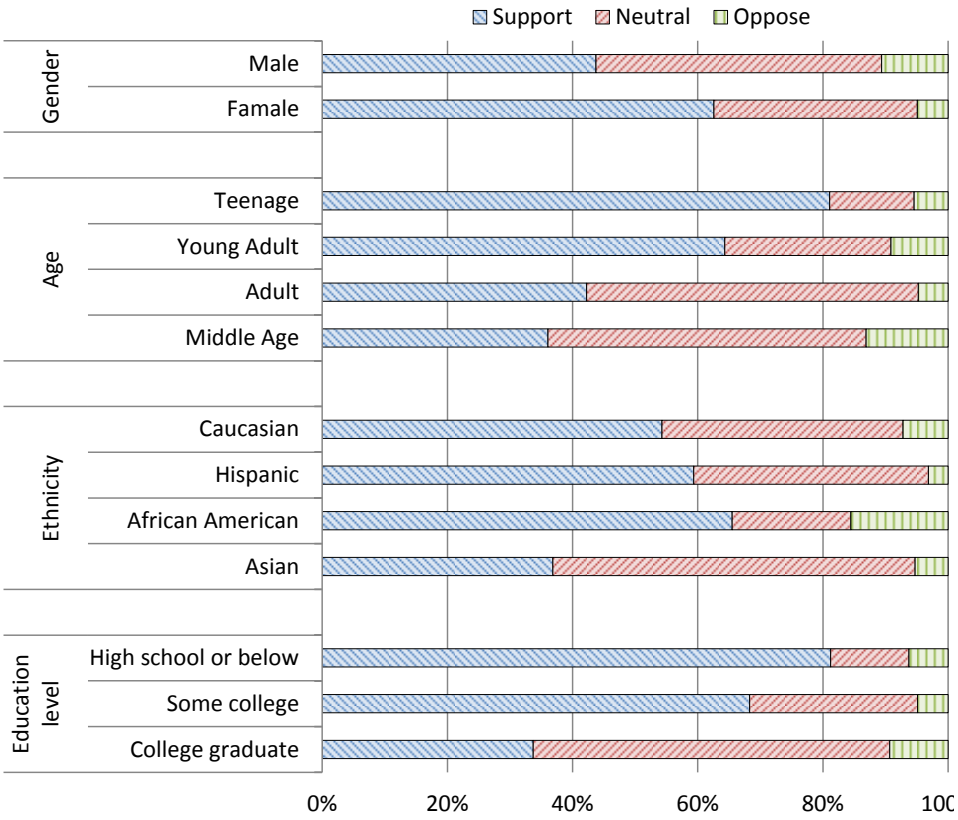

Figure 5-9: Support for same-sex marriage across different demographic categories.

servative characteristics were least supportive. Groups based on demographic attributes can similarly distinguish the opinions for the most common opinion (supportive) but fails to capture the opposition opinion. In contrast, our latent user characteristics do well (at least better than demographic attributes) in grouping both majority and minority opinions.

5.4 Implications

We have demonstrated that inferred user characteristics based on user self-descriptions can provide intuitive characteristics that aid in the analysis of political topics and form coherent opinion groups. In our three sets of experiments, we have consistently seen that

these inferred characteristics are more informative than traditional demographic groups, even when those groups are provided directly by human annotators. Furthermore, while not every user has an informative self-description, we found that the descriptions from the social network provide a more informative representation for the user. This conclusion supports previous findings that a user's friends in a social network can be just as informative, if not more so, of the user themselves [76]. Additionally, we believe we are the first to utilize user self-descriptions for mining information about users.

Our work suggests that better models of user characteristics might further improve our results. Our goal was to demonstrate the basic effectiveness of inferred characteristics. We have done so based on the user and her social network. Yet we have seen that the content of a user's messages provides other, though weaker, information. Models that combined all of these information sources into a single inference problem could produce better models. Additionally, there may be opportunities to better utilize background and analysis specific datasets in learning a single model. Finally, while our models improve over demographic information, we see room for incorporating demographics from traditional supervised classifiers into characteristic models as well. We leave the direction of improved modeling to future work.

While social media offers a variety of opportunities for political and social scientists to study public opinion, social media data lacks the type of demographics available in opinion surveys. Therefore, a significant amount of work has focused on learning demographic classifiers for Twitter. However, this relies on supervised classification algorithms which necessitate training data. These methods often do not generalize to new domains or populations. In contrast, our methods are unsupervised and can be re-estimated for any dataset of interest. This reduces the data resource requirements of new analyses. Furthermore, our more refined characteristics open the door to new types of analyses, even beyond what is available based on demographic questions in survey data. The result is an analysis that benefits from the traditional advantages of social media –

rapid data collection, large datasets – and provides a finer grained demographic analysis than traditional methods.

5.5 Summary

In this chapter, we first define and measure sentiment dynamics of political topics. We then propose an unsupervised learning model to infer user characteristics from Twitter users' profiles (self-descriptions). To the best of our knowledge, this is the first attempt to model the short and sometimes cryptic user profiles in a principle way. We collect millions of users and compare latent user characteristics against established demographic attributes in estimating sentiment of two popular political issues. By leveraging social network information, our automatically inferred user characteristics significantly outperform demographic attributes, which have been widely used to estimate public opinion in previous work. User characteristics found via our unsupervised method can divide otherwise mixed and blurred overall sentiment into coherent user opinion groups. This technique could help understand concerns and excitement from different types of users and capture their sentiment dynamics.

Overall, understanding users' characteristics is essential to more accurately model sentiment and capture its dynamics. The techniques introduced in this chapter improve sentiment analysis by modeling such characteristics revealed in their profiles, which allows us to observe and perform deep analysis of opinion changes different types of users.

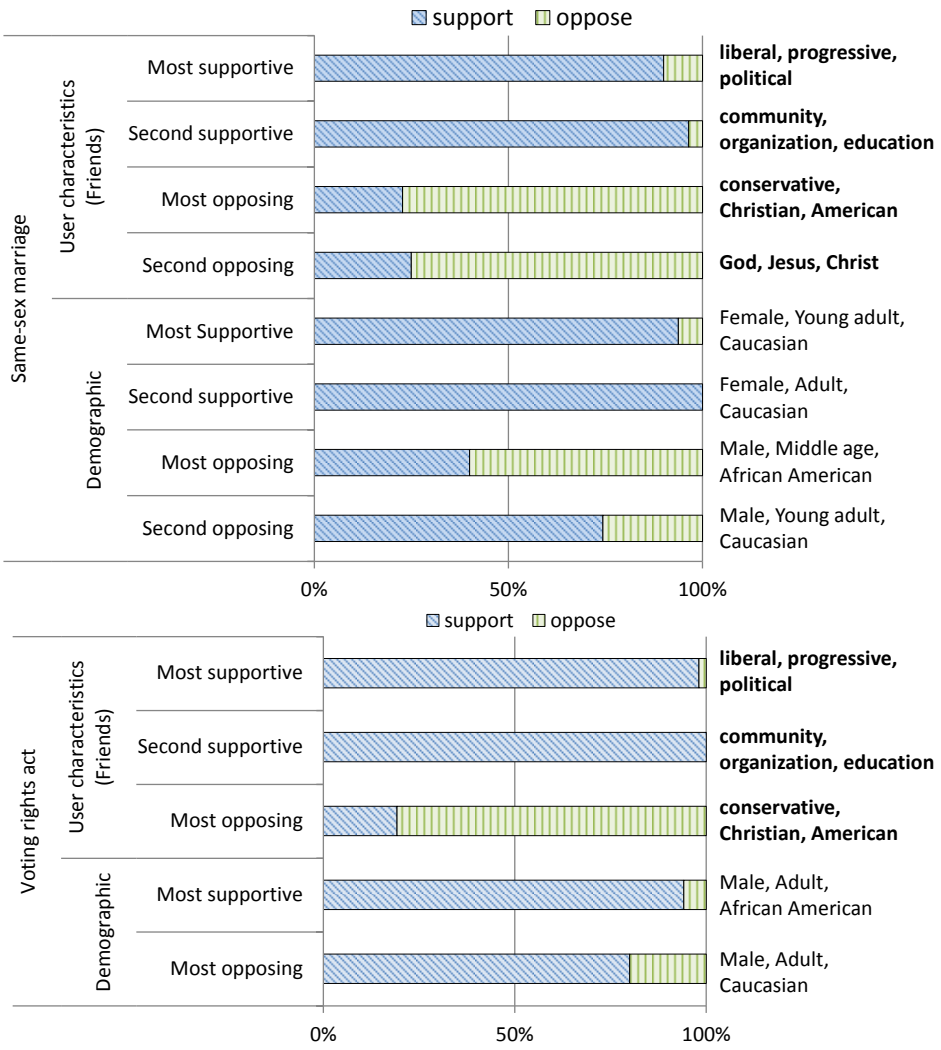

Figure 5-10: Top supportive and top oppositional groups by demographic information and user characteristics for same-sex marriage (left) and voting rights act (right). Text in the right column describes the groups: demographic labels and top three words for user characteristics).

Chapter 6

Modeling Temporal-Dependent User Behavior in Microblogging Content

Users' actions at a specific timestamp or in a particular temporal order could imply valuable information about the user and his preference. In this chapter, we study the relationship between time and two major types of user behavior: posting and following. For posting behavior, we develop a novel user characteristic inference model, UserTime, which incorporates timestamps of user's microblogging posts (when they tweet) into characteristic inference. The resulting latent characteristics of UserTime model improve profile completion and sentiment classification over the technique introduced in Chapter 5. For following behavior, we analyze the temporal order of link creation. The results indicate that early created links have better correlation with users' retweeting behavior than overall following probability, which implies that early links can better reflect users' information preference.

6.1 Modeling Time of Users' Posts to Improve User Characteristic Inference

The action of posting messages about an event or a trending topic could possibly reveal user's interests and characteristics. In this section, we model the timestamps of user's posts and incorporate this temporal information into the characteristic inference algorithm. Our proposed model, UserTime, leverage both users' profile words and the time of their Tweets to infer most robust user characteristics, which is especially valuable to overcome the challenge of data sparsity (i.e., when users have incomplete or empty profiles) in microblogging data.

6.1.1 Motivation

Users tend to comment on or spread news about popular events on Twitter. In fact, important events often draw users' participation and trigger topic and sentiment change in social media. In Chapter 5, we have shown how Supreme Court decisions have made millions of reactions on Twitter within a single day. Meanwhile, users usually have preference to different events based on their characteristics, such as location, belief, interests, etc. Thus, users' participation in different events could potentially reveal users' hidden characteristics to some extend. Such event-related participation in social media is often time-sensitive and occurs closely to corresponding events in the timeline. In Figure 6-1, we show participation of different user characteristics under the general topic "same-sex marriage" over time. The participation rates are quite different among users with different characteristics and in the temporal dimension. For example, when UK was discussing the issue of gay marriage, "british" users were more active than usual and other user groups. When U.S. Supreme Court decision and Britain policy favor same-sex marriage, enthusiasts have much higher participation rate than normal. A supportive Tweet from celebrity Lady Gaga triggered a burst of Tweets from "Lady Gaga

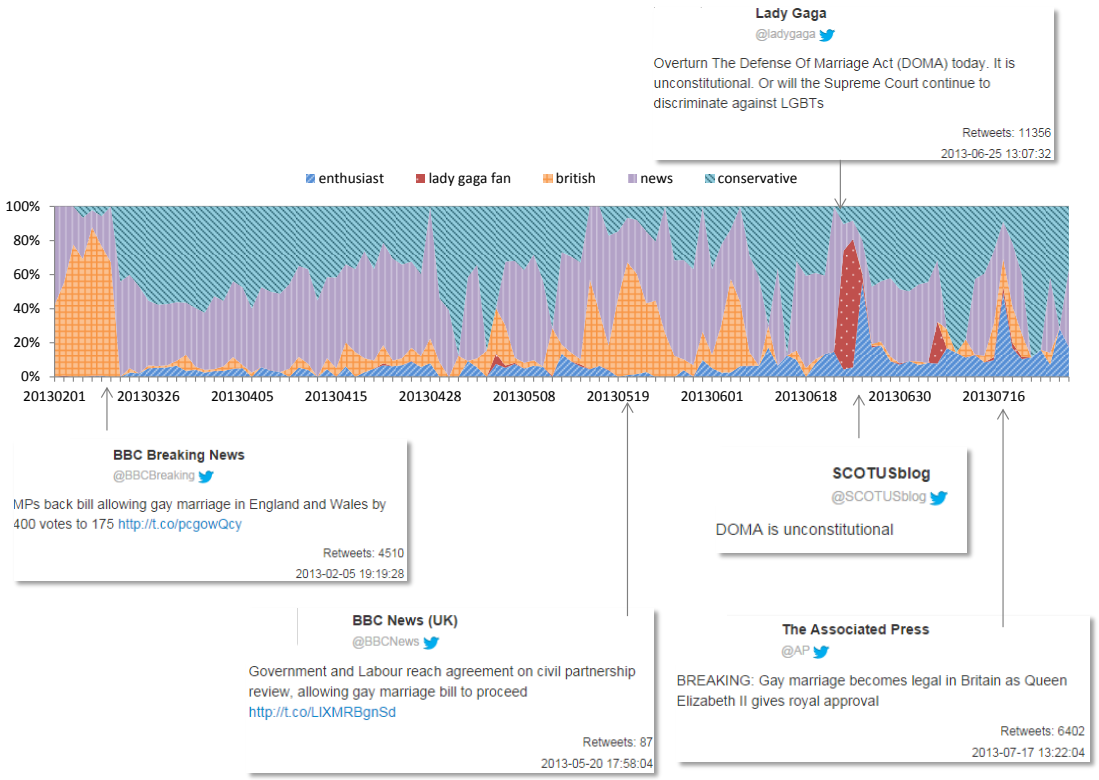

Figure 6-1: Users with different characteristics have distinct participation rates over time

fans". These observations imply that users with different characteristics have preferences on participating discussions of different events, and therefore users' participation rate over time may, to some extent, reveal their characteristics.

Statistically, if a user posts a Tweet about "same-sex marriage" on the day of Lady Gaga's Tweet, he probably belongs to "lady gaga fans" (from Figure 6-1) even his profile only mentions "music". Based on these observations and intuitions, we propose a user characteristic inference model which leverages both words in user's profile and the timestamp of his tweets under generic topics.

Our previous user characteristic inference model considers the text of users' descriptions is generated by latent characteristics. The automatically discovered characteristics show better performance in estimating political sentiment than human labeled demo-

graphic attributes. Now we want to integrate time of users' Tweets into the inference. Existing papers extend LDA and simultaneously model document content and timestamps, including dynamic topic models [14] and topics over time model [86]. The major technical differences between our proposed model and existing models are twofold:

- In our settings, "documents" are users' self-descriptions, and "timestamps" are associated with users' Tweets. These two elements are not directly connected, but through the authorship. Other models mostly model content and timestamps of the same documents.

- A document in previous work usually has one timestamp (the time when the document is created). However, since timestamps in our model do not directly describe the documents (self-descriptions), a document could have one or more timestamps.

Modeling and combining multiple timestamps with a single user's self-description brings challenges. First, we do not want to overweight users who tweet more often than others. For example, "news" and "conservative" users are persistently present under "gay marriage" topic. The high and persistent volume of their Tweets reflects a pattern of participation, which may or may not correlate with users' importance to political topics. Second, each user characteristic could have one or more bursts over time, and the bursts can be spontaneous, which makes existing model (e.g., topics over time [86]) difficult to fit since they use smooth (beta) probability function to describe the time distribution of topics or latent characteristics. Third, efficiency could be a problem when introducing additional factors (timestamps) into generative models.

The goal of modeling timestamps of users' posts is to overcome the challenges brought by data sparsity in Twitter users' self-descriptions. The short descriptions make it difficult to comprehensively reveal user's characteristics and segment user groups. Meanwhile, "when to post" also contains information about users' interests. The combi-

nation of profile words and Tweet timestamps could potentially make them complement each other and produce more robust user characteristics.

We propose a generative model, namely UserTime, to infer latent user characteristics by leveraging both words in users' self-descriptions and the timestamps of their Tweets. In this work, we reduce the time resolution to days, and consider days as tokens in another vocabulary space. The distribution over characteristics of a single user not only generates words in his profile, also generates which day(s) the user will tweet about a certain topic.

6.1.2 UserTime Model

Table 6.1 shows variables and their descriptions used in the following discussions.

Variable	Description
$\boldsymbol{W}$	all words in users' profile
$\boldsymbol{V}$	vocabulary of all users' profile
$\boldsymbol{X}$	unique days of all users' Tweets
$\boldsymbol{T}$	all days of users' Tweets
$\boldsymbol{Z}$	all user characteristics
α, β, γ	priors of corresponding Dirichlet distributions
ψ_z	day distribution of characteristic z
φ_z	word distribution of characteristic z
θ_u	characteristic distribution of user u
$z_{u,w}$	characteristic assignment of word w in user u's profile
$z_{u,t}$	characteristic assignment of day t of user u's Tweets
$u(w)$	all words in u's profile
$u(t)$	all days of u's Tweets
$nw^z_{u,\cdot}$	number of words in u's profile assigned to z
$nw^z_{\cdot,v}$	number of times word v assigned z across all users
$nt^z_{u,\cdot}$	number of days in u's Tweet days assigned to z
$nt^z_{\cdot,x}$	number of times day x assigned z across all users

Table 6.1: Notations of UserTime model.

Figure 6-2 describes the generative process of UserTime model.

For each user u:

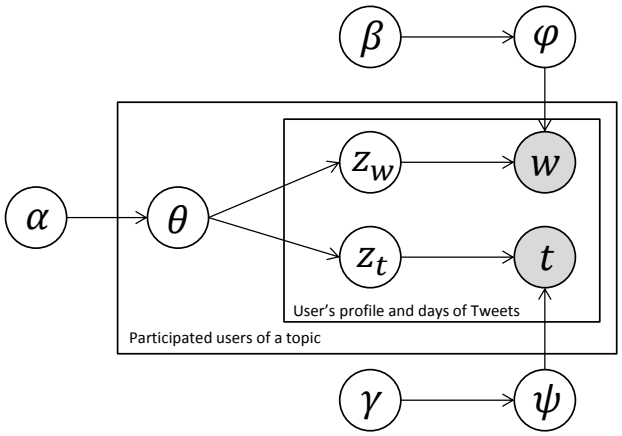

Figure 6-2: Plate notation of UserTime model.

- Draw a multinomial θ_u from Dirichlet distribution for parameter α. $\theta_u \propto Dir(\alpha)$
- Draw word-characteristic distribution φ from $\varphi \propto Dir(\beta)$
- Draw day-characteristic distribution ψ from $\psi \propto Dir(\gamma)$
- For each word in u's profile:
  - Draw a characteristic z_w from θ_u
  - Draw a word w from φ_{z_w}
- For each day u authored any Tweets:
  - Draw a characteristic z_t from θ_u
  - Draw a day t from φ_{z_t}

In the generative process, words and timestamps of Tweets are simultaneously generated from latent characteristics. When inferring characteristic distribution θ_u, both of them will be taken into account.

We adapt Gibbs sampling to perform the inference. According to the model, the total probability to generate the whole corpus, including words and time, is shown in

Formula 6.1.

$$
\begin{aligned}
&P(\boldsymbol{W}, \boldsymbol{T}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{\psi} | \alpha, \beta, \gamma) \\
&= \prod_{z \in \boldsymbol{Z}} P(\varphi_z|\beta) P(\psi_z|\gamma) \prod_{u \in D} P(\theta_u|\alpha) \prod_{w \in u(w)} P(z_{u,w}|\theta_u) P(w|\varphi_{z_{u,w}}) \prod_{t \in u(t)} P(z_{u,t}|\theta_u) P(t|\psi_{z_{u,t}})
\end{aligned}
\tag{6.1}
$$

As shown the in Formula 6.1, the total probability consists of three major parts: generating characteristic distribution θ_u, generating the profile words, and generating the timestamps of Tweets. The first two parts are the same as LDA. The third part, the probability of generating Tweet timestamps, is the key ingredient in UserTime model.

To perform Gibbs sampling, we need to integrated out θ, φ and ψ. The overall probability becomes Formula 6.2.

$$
\begin{aligned}
&P(\boldsymbol{W}, \boldsymbol{T}, \boldsymbol{Z} | \alpha, \beta, \gamma) \\
&= \int_{\theta} \prod_{u \in D} P(\theta_u|\alpha) \prod_{w \in u(w)} P(z_{u,w}|\theta_u) \prod_{t \in u(t)} P(z_{u,t}|\theta_u) d\theta \\
&\quad \times \int_{\varphi} \prod_{z \in \boldsymbol{Z}} P(\varphi_z|\beta) \prod_{u \in U} \prod_{w \in u(w)} P(w|\varphi_{z_{u,w}}) d\varphi \\
&\quad \times \int_{\psi} \prod_{z \in \boldsymbol{Z}} P(\psi_z|\gamma) \prod_{u \in U} \prod_{t \in u(t)} P(t|\psi_{z_{u,t}}) d\psi \\
&= \prod_{u \in U} \frac{\Gamma(\sum_{z \in Z} \alpha_z)}{\prod_{z \in Z} \Gamma(\alpha_z)} \frac{\sum_{z \in Z} \Gamma(nw^z_{u,\cdot} + nt^z_{u,\cdot} + \alpha_z)}{\Gamma(\sum_{z \in Z} nw^z_{u,\cdot} + nt^z_{u,\cdot} + \alpha_z)} \\
&\quad \times \prod_{z \in Z} \frac{\Gamma(\sum_{v \in V} \beta_v)}{\prod_{v \in V} \Gamma(\beta_v)} \frac{\sum_{v \in V} \Gamma(nw^z_{\cdot,v} + \beta_v)}{\Gamma(\sum_{v \in V} nw^z_{\cdot,v} + \beta_v)} \\
&\quad \times \prod_{z \in Z} \frac{\Gamma(\sum_{x \in X} \gamma_x)}{\prod_{x \in X} \Gamma(\gamma_x)} \frac{\sum_{x \in X} \Gamma(nt^z_{\cdot,x} + \gamma_x)}{\Gamma(\sum_{x \in X} nt^z_{\cdot,x} + \gamma_x)}
\end{aligned}
\tag{6.2}
$$

The resulting formula has three components: characteristic distribution θ_u, profile words, and Tweet timestamps. The last two components have the same form, which is

the likelihood of generating profile words and Tweet timestamps respectively. The first component illustrates that how UserTime model use Tweet timestamps as a complement to infer user characteristics: both words and timestamps assigned to each characteristic are taken into account in estimating θ_u. The form is sum, which means each observation (word and time) has equal weight and the observation of Tweet timestamps could contribute to inference when the profile words are insufficient.

Since the generative process in UserTime has two parts: generating words in user's profile, and generating the timestamps of user's Tweets. The actual Gibbs sampling will take turns. When assigning characteristics to word, the probability of word w in user u's profile is assigned characteristic z will be as Formula 6.3.

$$P(z_{uw} = z|\boldsymbol{W}, \boldsymbol{T}, \boldsymbol{Z_{-uw}}, \alpha, \beta, \gamma) \propto (nw_{u,\cdot}^{z,-(u,w)} + nt_{u,\cdot}^{z} + \alpha_z) \times \frac{nw_{\cdot,w}^{z,-(u,w)} + \beta_w}{\sum_{v \in V} nw_{\cdot,v}^{z,-(u,w)} + \beta_v} \tag{6.3}$$

When assigning characteristics to the days of Tweets, the probability of day t of user u's Tweets is assigned characteristic z will be as Formula 6.4.

$$P(z_{ut} = z|\boldsymbol{W}, \boldsymbol{T}, \boldsymbol{Z_{-ut}}, \alpha, \beta, \gamma) \propto (nt_{u,\cdot}^{z,-(u,t)} + nw_{u,\cdot}^{z} + \alpha_z) \times \frac{nt_{\cdot,t}^{z,-(u,t)} + \beta_t}{\sum_{x \in X} nt_{\cdot,x}^{z,-(u,t)} + \beta_x} \tag{6.4}$$

In our setting, we only consider unique days for a user's Tweets. In other words, the user will have day x only once even if he posted more than one Tweets on that day. By doing so, we attempt to avoid overweighing the time component. From Formula 6.3 and 6.4, we can see that days are as important as the actual words in user's profile. However, intuitively, time of Tweets is a weaker signal of user's characteristics. We want to use it as a complimentary in the inference. By transforming Tweets to unique days, UserTime model is less sensitive to number of Tweets and more robust to infer user characteristics.

The results of UserTime model consist of three major parts, ψ_z, φ_z, and θ_u. ψ_z and φ_z can be computed by following the convention of LDA. And θ_u, characteristic distribution of user u, can be estimated as follows:

$$\hat{\theta_u^z} = \frac{nw_{u,\cdot}^z + nw_{u,\cdot}^z + \alpha_z}{\sum_{z' \in Z} nw_{u,\cdot}^{z'} + nw_{u,\cdot}^{z'} + \alpha_{z'}}$$

6.1.3 Experiments and Results

In this section, UserTime model is evaluated against baseline model LDA. In Chapter 5, we showed the user characteristics inferred by LDA and the dataset used for evaluation. We apply UserTime model on the exact same dataset and compare against LDA. Note that the dataset is not restricted by demographic attributes any more, so that we can run our experiments on the full set of users, which leads to slightly different experimental results in sentiment classification. Details about the data collection are stated in Section 5.3.3. We evaluate UserTime with three types of experiments: (1) Harmonic mean of generative probability of held-out dataset. (2) Generative likelihood of held-out words in users' profile. (3) Using the result to classify user's sentiment (support) of political topics.

Harmonic Mean

Harmonic mean method has been widely used in evaluating topic models [36, 38, 82] due to the fact that it can be naturally implemented with results of Gibbs sampling. When compute the harmonic mean of generative likelihood of held-out dataset, we use importance sampling [83]. Formula for computing harmonic mean is as follows:

$$P(\boldsymbol{W}|\varphi, \alpha) \simeq \frac{1}{\frac{1}{|\boldsymbol{W}|} \sum_{w \in \boldsymbol{W}} \frac{1}{P(w|z_w, \varphi)}}$$

The formula above gives average (harmonic mean) probability of generating held-

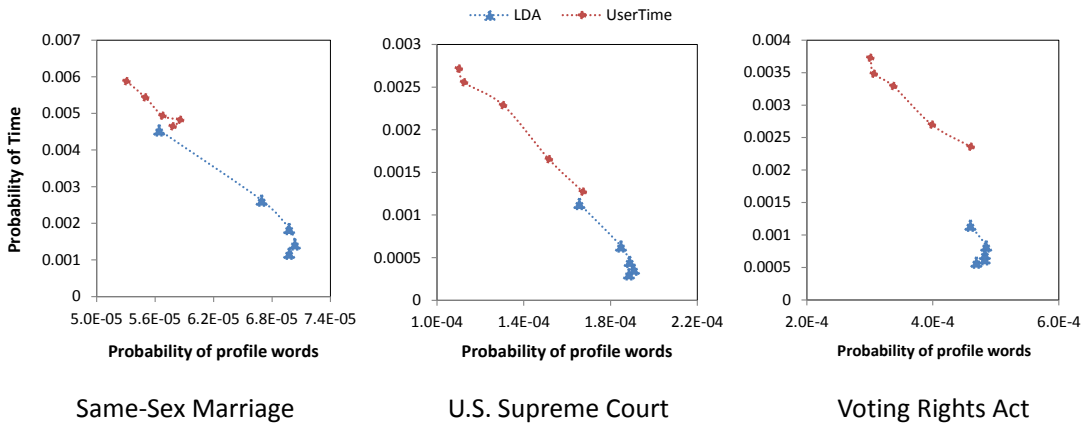

Figure 6-3: Harmonic Mean: probability of generating profile words and timestamps (days) of Tweets

out dataset from a trained model. Since, UserTime model is capable of generating both profile words and time of Tweets, we compute the harmonic mean for both words and time. In LDA, there is no characteristic assignment for time of Tweets. We tweak the process to let LDA generate time of Tweets via θ_u. That is, for each unique day of the Tweets in the hold-out dataset, we randomly draw a characteristic from multinomial distribution θ_u. And then the time distributions over characteristics are obtained. Therefore we can use the resulting time distributions to estimate generative probabilities for days of tweets, making LDA comparable to UserTime model.

Since both models could generate words and time after modifications mentioned above, we plot the probabilities on a two dimensional figure where horizontal axis is probability of words and vertical axis is probability of days.

After running both models with different number of characteristics (100, 200, ..., 500), we present the results in Figure 6-3. Intuitively, the model with higher probability on both axis (upper right of the chart) can be considered better than the one on the lower left corner. It turns out that UserTime and LDA produce very different results: LDA focus on words only, producing higher likelihood for profile words. Whereas UserTime model generates time with much higher probability. Overall, it is difficult to judge

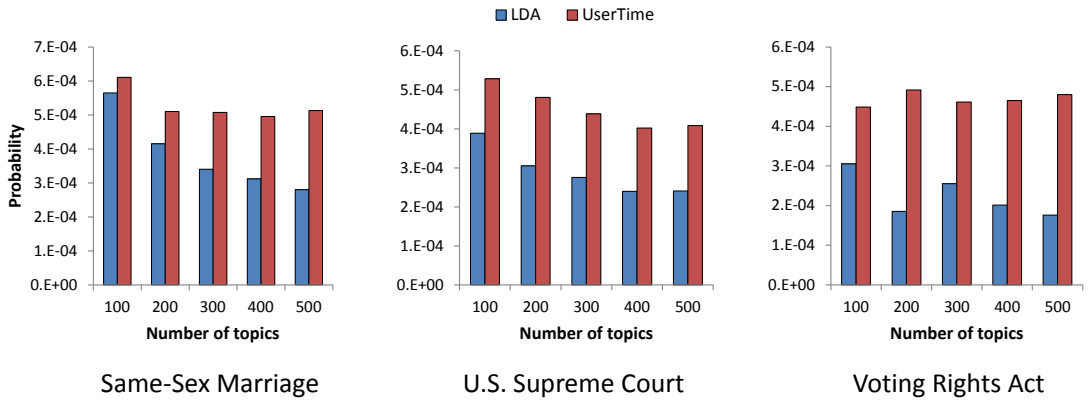

Figure 6-4: Profile Completion: probability of generating unrevealed words in user profiles

which one performs better. UserTime model is slightly positioned towards the upper right corner of the chart.

Profile Completion

One of our claims is that time of user's tweets can reveal his missing characteristics from his profile words. We now turn to the task of recovering held-out profile words. This evaluation method was used in author-topic model [74]. The idea is that for each held-out document (a user's self-description), we hide part of it and only infer cluster assignments on the other part. After that, we "guess" the hidden part through characteristics distribution inferred by the visible part. This is a simplified task of profile completion which has the potential to facilitate application such as profile word suggestion and users' hidden attributes estimation. In our setting, we hide one random word from each held-out user profile, and report average generative probability of the hidden words. The baseline model, LDA, only use the visible words from user's profile to infer the hidden words. In contrast, UserTime model uses both the visible words and time of user's Tweets to perform the inference.

Figure 6-4 shows the averaged probabilities of generating hidden profile words of

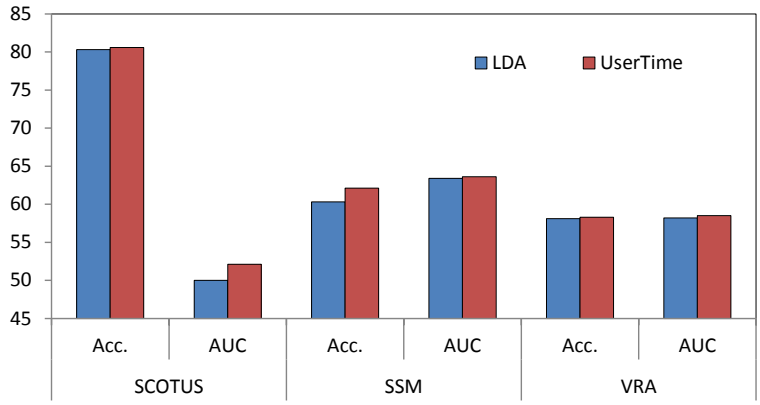

Figure 6-5: Performance of user sentiment classification based on 10-fold cross-validation. SCOTUS is U.S. Supreme Court, SSM is same-sex marriage, and VRA is voting rights act.

two models. UserTime model consistently outperforms in recovering missing profile words. Time of user's Tweets, the additional signal that UserTime has, is helpful in inferring user's characteristics. This task, selecting proper words from the whole dictionary to fill into user's profile, is challenging for both models and the probability tends to be small.

Sentiment Classification

We want to see whether the characteristics inferred by UserTime model can improve downstream applications over the baseline model. The application is exactly the same as before: Users were classified as either being supportive of, neutral towards, or in opposition to the political topic "same-sex marriage", "voting rights act", and "U.S. Supreme Court". "U.S. Supreme Court" is additionally added to the dataset for evaluating inferred characteristics with a more generic topic. The experimental setup and data collection are also the same as before. The only difference is that we are using characteristic weights from UserTime model instead of standard LDA.

Figure 6-5 shows that UserTime model slightly but consistently improves the accuracy and AUC metrics over the baseline.

6.1.4 Discussion

UserTime model integrates both user's profile and time of Tweets into characteristics inference. And the resulting characteristics outperform the ones inferred by LDA in both profile completion and sentiment classification. It also generates time-characteristic probabilities ψ, which can be used to understand the active periods for different user characteristics.

Users on Twitter often have short profiles, which is nearly impossible to reveal their comprehensive characteristics. Previously, we showed that using social network information can improve the robustness of user characteristic inference. However, obtain user's social network is costly through Twitter API, and this method could not work well if the user does not have many connections. In this work, our UserTime model illustrates a possibility of using temporal-dependent behavior to improve characteristic inference and recover user's hidden characteristics. Future work in this direction would include leveraging not only the time of the Tweets, but the actual content of what users say about relevant events.

6.2 Modeling Temporal Order of Users' Followings to Measure Link Importance

User's social network is often constructed over time. The temporal order of establishing following links may contain valuable information about links themselves. Intuitively, early created links may be more important to the user since they stand out among users' selection of followings in time dimension. In the setting of Twitter, retweet could only happen if there is a link between the retweeter and retweetee. Retweets could represent users' attention, information consumption, and even opinion agreement along the links. In this study, we use number of retweets on a characteristic-level to represent link im-

portance. We extend the user characteristic inference technique introduced in Chapter 5 and take three steps to validate this hypothesis: (1) build temporal following preference matrix for user characteristic group; (2) build retweet preference matrix for user characteristic group; (3) compute the correlation between them.

6.2.1 Retweets as A Measure of Link Importance

As one of the most heavily used feature, retweeting friends' posts can reflect retweeters' interests and content consumption of those messages. The analysis in Chapter 5 shows that retweets compose a large portion of the sentimental Twitter messages. Highly retweeted messages often represent interesting and popular opinions and sentiment. Retweeting a post could even imply users' agreement of the opinions in the message. Thus, investigating user's preferred sources of retweets could help understand how sentiment and opinions spread within and across user characteristic groups. In Twitter, retweets are restricted to users with connections in between, which makes the network structure an important factor in studying retweeting behavior. The following relationship defines how the content or information could flow in Twitter and many other social media sites. Moreover, in previous sections, experiments of inferring users' characteristics from profiles shows that social network information (who they follow on Twitter) can greatly improve the quality and robustness of the inference, which in turn produces the best results in sentiment classification. In this section, we study the correlation between "whom they follow" and "whom they retweet from". Our results suggest that the temporal order of establishing the following relationship provides better signal of predicting retweet behavior on characteristic-level than the aggregated following probabilities which ignore the temporal information.

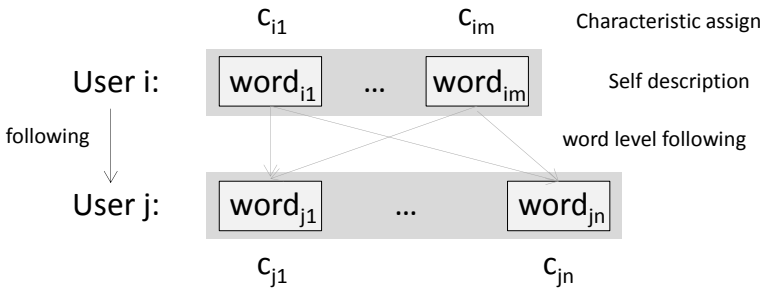

Figure 6-6: Constructing characteristic-level following relationship from user-level following information.

6.2.2 Temporal Following Preference across User Characteristics

From the result of user characteristic inference, we have characteristic assignment for each word in user's self-description. Suppose we have user i following user j, and characteristic assignment of words in their profile available, a characteristic-level following relationship can be constructed as in Figure 6-6.

First, we break down the user-level following links to word-level. Directed links are created from each word in user i's profile to each in user j's. If the length (number of words) of i's profile is m and length of j is n, there would be $m \times n$ links from i to j. Each word-level link can be considered as a characteristic-level link since every word has its characteristic assignment. However, we do not want to overweight users with long profile, meaning that every user-level following link should contribute equal amount of credit to characteristic-level following weights. Thus, the weight of each word-level link should be normalized by the total number of links generated from user-level followings. In the case of Figure 6-6, each word-level link would have the weight of $\frac{1}{m \times n}$.

After aggregating all user-level links in the dataset and translating them into characteristic-level following information, a characteristic following matrix is built as shown in Figure 6-7. Note that the matrix is row-wise normalized to reflect preference of following of each user cluster.

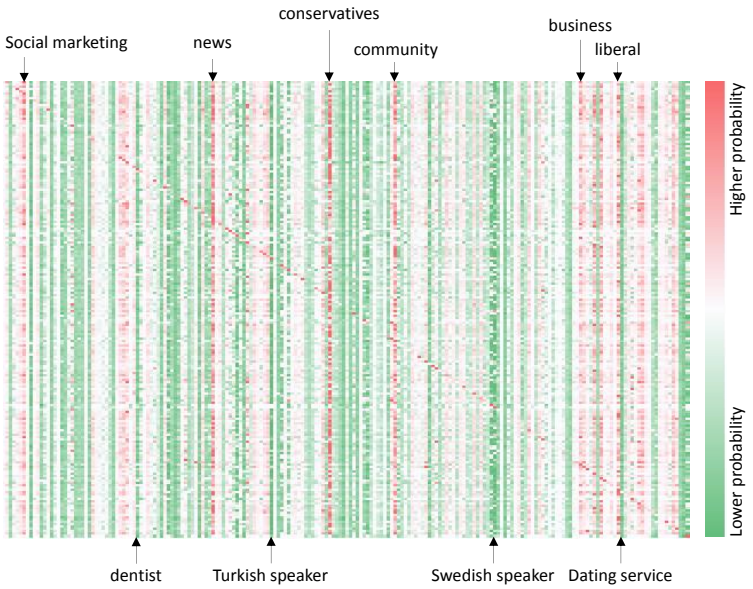

Figure 6-7: Characteristic-level following probability, entry (i, j) indicates the probability of users with characteristic i (row) following ones with characteristic j (column).

Figure 6-7 visualizes the characteristic-level following probability. The clear strong diagonal line in the figure indicates users tend to follow the ones who share their own characteristics. In social science, it refers to the phenomenon of "homophily": it's easier for similar users to establish bonds. We also see some vertical patterns: some columns are very strong (in red) and some others are very weak (in green). The "red" columns represent popular characteristics to follow, including "news", "conservatives", "business", etc. And the green columns are less popular, and many of them are non-English speakers.

6.2.3 Retweeting Preference for Users with Different Characteristics

Tweets collected via Twitter API contain a field called "retweeted status" if it is retweeted from friends. Also, we know the profile of both the original author and the one who

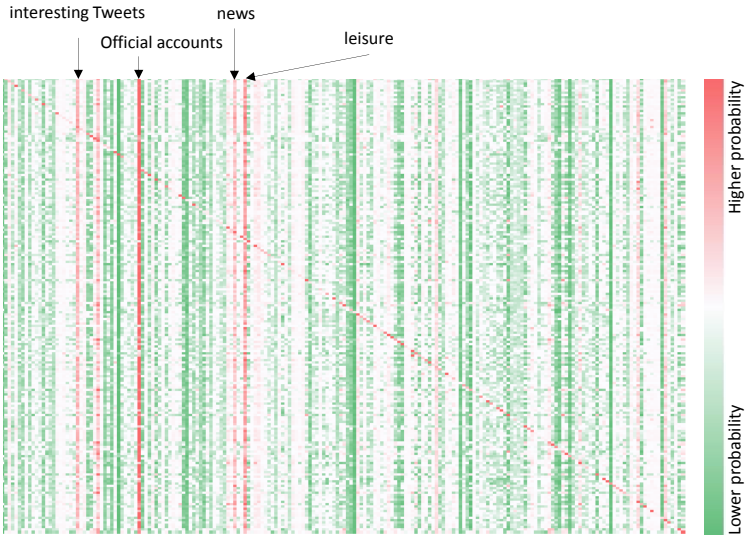

Figure 6-8: Characteristic-level retweeting probability, entry (i, j) indicates the probability of characteristic group i (row) retweeting from characteristic group j (column).

retweeted the message. Note that these two users may not directly connected in the network. However, it still means that a message from original user is delivered to and selected by the retweeter. The construction of retweeting preference matrix is quite similar as the following matrix: we assign user characteristics to each token in the two users' profiles, and a retweet relationship is translated into characteristic-level retweeting weight by a similar method as in Figure 6-6. Suppose we have 200 characteristics in total, a 200-by-200 retweeting matrix can be built. Entry (i, j) records the weight or probability of retweets authored by characteristic group j (column) and retweeted by characteristic group i (row). When the matrix is normalized row-wise, we have the distribution of retweeting preference for each user characteristic.

Figure 6-8 visualizes the retweeting probability. Similar as the following probability matrix, the diagonal line is very strong, which implies that users retweet more from the ones with similar characteristics. The red vertical lines indicate popular sources of retweets, including accounts dedicated to generate or promote "interesting Tweets",

official and verified accounts of organizations, news, etc.

6.2.4 Correlations between Retweets and Followings

We first look at the correlation between following probability and number of retweets for each user characteristic. For each characteristic i, we compute Pearson's correlation coefficient between the ith row of following matrix and ith row of retweet matrix. Each of the vectors consists of 200 numbers which represents the preference of characteristic i over all 200 characteristics. The averaged correlation of all 200 user characteristics is 0.41, and the average p-value is 0.001. It seems these two factors, following and retweet, fairly correlated with each other. This is not surprising since following reflects users' interests and so does retweeting.

Second, we compute the following probability within each bucket of temporal index when the links are established. For example, we calculate the likelihood of users with characteristic i following the ones with characteristic j within the first 100 links that users of i initiated. We do so for 19 buckets and each bucket has the size of 100. In other words, we calculate the following probability within the first 100 links, the second 100 links, and till the 19th 100 links. The last bucket consists of all the links which are created later than the 1900th link. The results are 20 following probability matrices, and each of them represent the following preference within the corresponding time frame. After that, we compute the correlation between each of the temporal following matrix and the retweeting preference matrix.

Figure 6-9 shows the Pearson's correlation coefficients between the following preference matrices and the retweeting matrix. The results indicate that early created links have higher correlations with number of retweets between user clusters. If we consider retweeting preference as an indication of link importance, this finding can be interpreted as "early created links are more important".

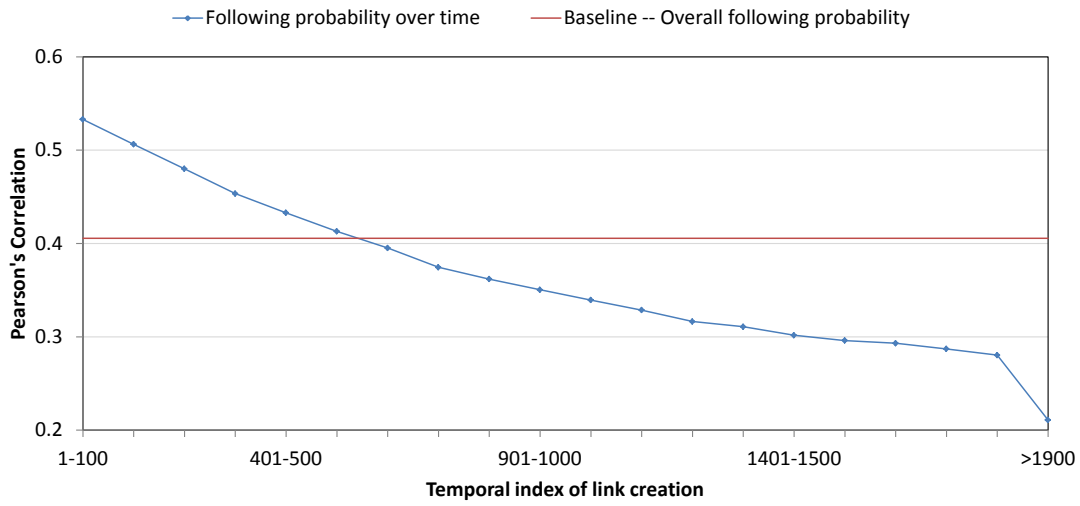

Figure 6-9: Pearson's correlation between number of retweets and two factors:(1) following probability in each temporal index bucket; (2) overall following probabilities.

6.2.5 Correlations between Early Followings and Overall Followings

Finally, we investigate the correlation between temporal following probability and the overall following probability. Figure 6-10 show the Pearson's correlation coefficients. Surprisingly, links created between 600th and 700th give the best correlation with the overall probability. Probability calculated from early created links (1st - 100th), however, has the lowest correlation. This results suggest that early created links may not provide better signals about following preference in the long term than later created ones, but they are more indicative for the link importance in terms of retweeting preference.

6.2.6 Implications for Sentiment Dynamics

Users often express their opinions and spread sentiment by retweeting posts from the ones they are following. Analysis in this section shows that early created following

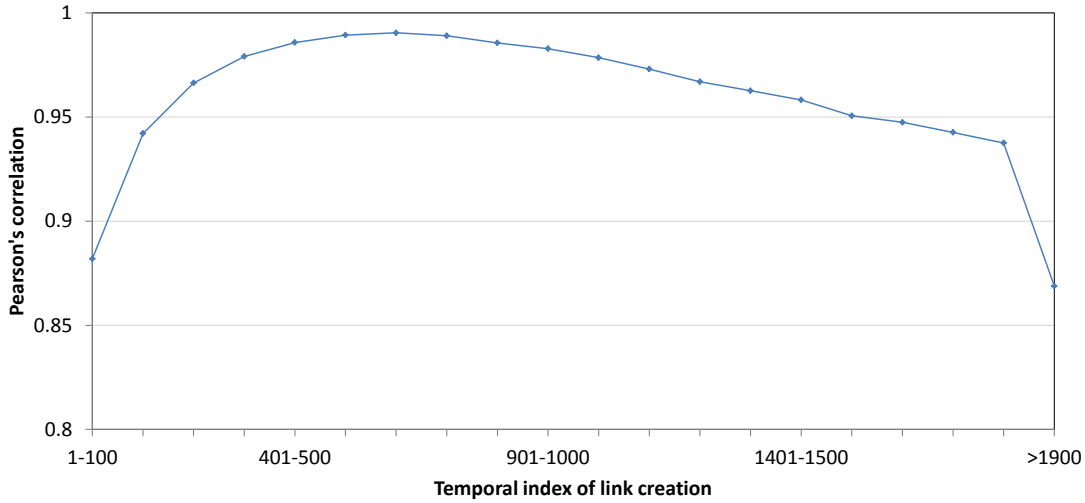

Figure 6-10: Pearson's correlation between overall following probabilities and following probability in each temporal index bucket.

relationship have stronger correlation with the retweeting behavior. This result suggest that temporal order of link creations in social networks could indicate how sentiment would propagate through the network. Also, this finding can be used to weight the links with their temporal information, which could potentially help better infer users' characteristics with a weighted following network.

6.3 Summary

In this chapter, we model temporal-dependent user behavior, including posting and following, to investigate information contained in the timestamps or temporal order of user actions. We improve the quality of inferred characteristics by incorporating timestamps of users' Tweets into the inference. The proposed model, UserTime, achieves better sentiment classification results than LDA, and outperforms LDA in the task of recovering hidden words in users' profiles. We adopt the user characteristic inference techniques developed in chapter 5 to examine the temporal order of link creation and retweeting behavior on characteristic group level. The correlation analysis shows that the tempo-

ral order of link creation in Twitter is a stronger indicator of retweeting among users with different characteristics than the overall following relationship. Early created links appear to more likely carry retweets, and therefore can provide more signals of users' preference of information sources. This observation suggests the way of weighting links with their temporal information in user characteristic inference, sentiment analysis, and network analysis tasks.

Chapter 7

Visualizing Temporal Dynamics in Social Media

In this chapter, we present a website named *courtometer.com* to visualize and analyze dynamics of political topics in microblogging content. The website is powered by the techniques introduced in Chapter 5 and runs sentiment classification and user characteristic inference in real-time. To aid researchers discover and analyze the dynamics, we develop several features to let users customize and build subtopics with additional keywords, author locations, and clustering users with inferred characteristics. The resulting trends and dynamics can be shared across the site and downloaded for offline analysis.

7.1 Motivation

As shown previously, measuring and modeling dynamics in sentiment and topics could produce interesting findings and patterns. As to the domain of political science, the political topic and sentiment dynamics on Twitter could help better understand the issues related to the cases before the Supreme Court âĂŞ which could have monumental and lasting importance on people's lives but may not receive sufficient exposition or

attention in popular media. Thus, it becomes critical to make the data and our measures available to public and let users explore and find their own trends of dynamics. We believe that the most promising way to do this is through the construction of a public website that will make available a wide range of data and also provide analytic and visualization tools for users. In this light, we build a website named *courtometer.com*, which visualizes the popularity and sentiment change of political topics on Twitter in realtime, and let users create additional customized filters to zoom into more specific type of content.

7.2 Goals and Features

We design *courtometer.com* with the following goals:

- Visualizing the trends: Topic and sentiment dynamics in Twitter can be considered as number of corresponding Tweets over time. We believe the most intuitive way to present the dynamics is to visualize the trends in graphics. This allows users to capture the burst of certain trend and play with the dynamics by simply interacting with the figure.

- Customizing the dynamics: Users with different needs may want to focus on different aspects of the dynamics. To serve as many users as possible, the system should be generic enough and meanwhile support user-defined customization of dynamics. Specifically, we want users to customize the trends by providing additional rules and restrictions, such as matching additional keywords, restricting author types or locations, etc.

- Understanding the dynamics: It is often a challenge to understand why dynamics occur at a certain time. Users need supporting context to truly understanding

Figure 7-1: Example political topics on *courtometer.com* for year 2014.

the potential cause of the dynamics. It is preferable to show top opinions when dynamics happen.

- Sharing the results: We want the findings on *courtometer.com* be easy to access and helpful for other research and applications.

To achieve these goals, the website, *courtometer.com* has 6 major features.

First, it visualizes the trends (the volume of Tweeting on our various topics) in a chart. The topics and the corresponding keywords (used to track relevant Tweet) are defined by political science experts. Example topics on the website are shown in Figure 7-1. We believe that these topics are popular and generic enough so that users can customize upon them.

Second, it allows users to select various sentiment measures. The available sentiment measures are “intense”, “happy”, “angry”, “support”, “oppose”. In Figure 7-2, we show the trend of the topic “civil liberties” and the trends of Tweets with corresponding sentiment leanings.

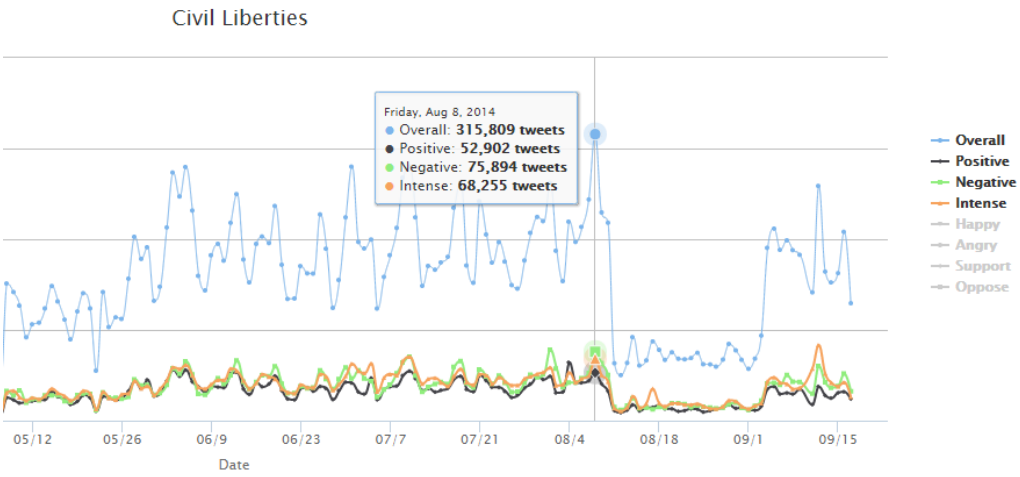

Figure 7-2: Trends of topic "Civil Liberties" and sentiment dynamics.

Third, the website allow individuals to create custom topics (as shown in Figure 7-3), namely subtopics, by specifying one or more of the following criteria:

- Keywords in the Tweets. All the resulting Tweets will contain at least one of the user-specified keywords.
- Location of the authors. The resulting Tweets are generated by users who claim their location to be the specified ones in their profiles.
- User clusters. The resulting Tweets are authored by users clustered by a certain characteristic. Example clusters include "business", "academic", "religious", etc.

The combinations of these three types of filtering can generate many interesting subtopics. For example, we can check how religious users in San Francisco react to the Supreme Court decision of same-sex marriage case; what business people say about President Obama with regard to his policies, etc.

Figure 7-4 shows an example subtopic under the general topic "civil liberties". The subtopic has the filter of keyword "Obama" and requires user cluster to be academic. The resulting trends are generated in real-time and show the popularity and sentiment change of the subtopic.

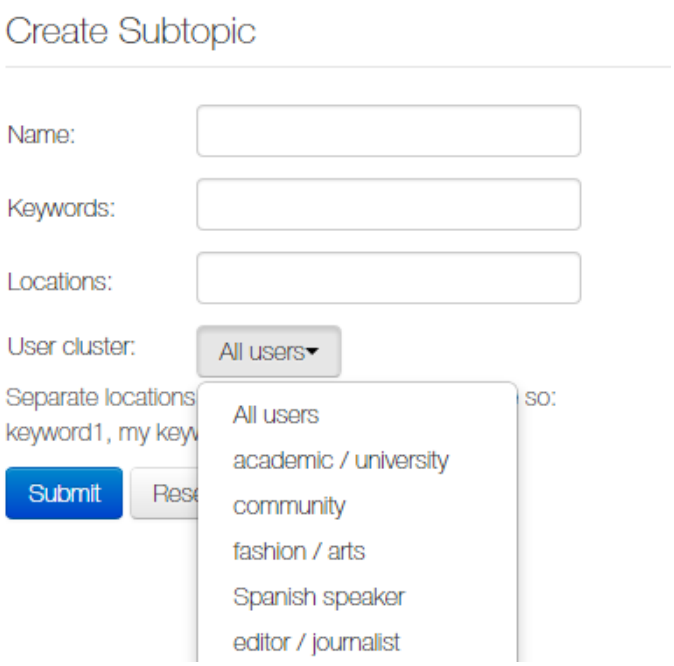

Figure 7-3: Interface of creating a subtopic.

Fourth, one can click on a particular day on the sentiment trends of both general topics and subtopics to see the top retweeted posts in different sentiment measures. The bursts on the sentiment curves could indicate a big sentiment change, but it cannot tell the context and what causes those changes. This "top tweets" feature is designed to reveal the most popular Tweets of that day, with the hope to show some hints about the context of the day and why such sentiment change occurred.

Figure 7-5 illustrates the "top tweets" feature: when the user click on a day (a dot on the line) of either a general topic or a customized subtopic, the panel will show the most retweeted Tweets of that day with corresponding sentiment. This figure shows top tweets of a subtopic created previously on the day August 8th, 2014.

Subtopics are created in "private" mode by default, meaning that only the owner can see the resulting topics. After the owner is happy with the findings, he can choose to publish subtopics and share them with other users or researchers on this platform. Users can download the findings to their local repository by exporting figures and raw

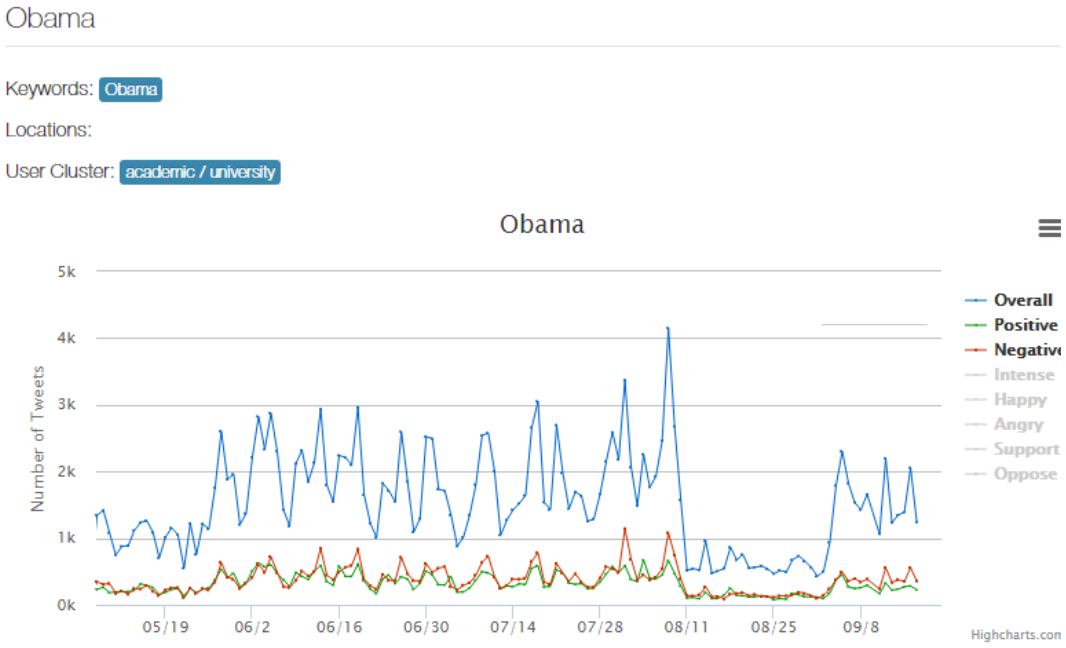

Figure 7-4: Sentiment trends of a subtopic.

data (sentiment counts, not the actual Tweets). With all these sharing and exporting features, data on *courtometer.com* is truly open and accessible to all users, and for many purposes.

Fifth, users can overlay trends from different topics, subtopics, and sentiment on the same chart. This feature allows users to visually observe and analyze potential correlations between dynamics. Figure 7-6 shows a chart that has the trends from "civil liberties" and "civil rights". The topics may have very different popularity. We support various ways to normalize the trends and compare the dynamics on a similar scale.

Sixth, users can share the figures they generated on *courtometer.com*. Once the figure is "published" by the user, other people will see that figure when they check out the same general topic. Also, users can download the figure and the data for offline analysis.

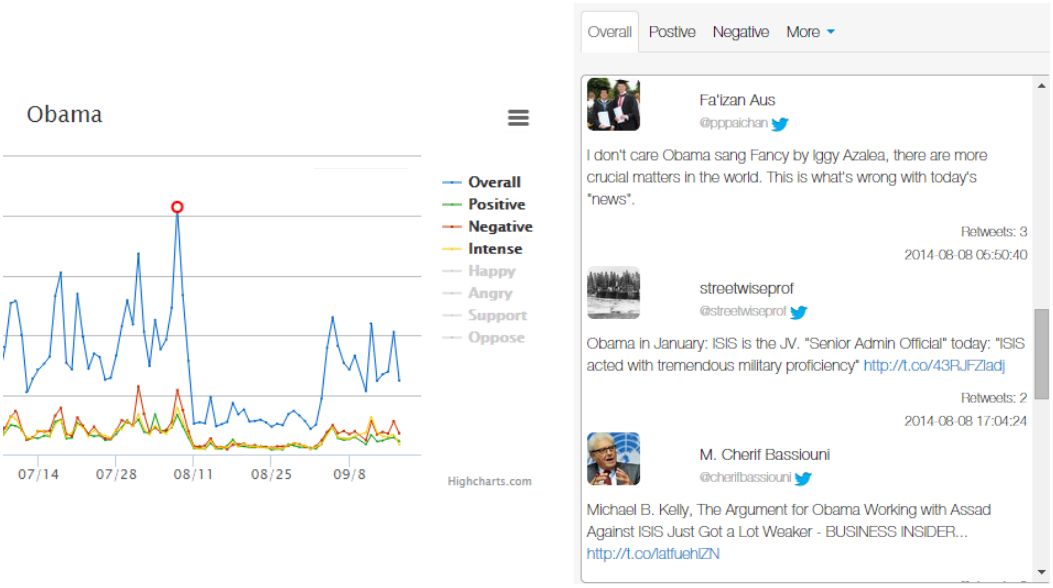

Figure 7-5: Top Tweets ranked by number of retweets for a subtopic

7.3 System Design

It is challenging to support features mentioned earlier, especially when most features can take place in realtime. Figure 7-7 shows the simplified structure of the supporting elements for the website.

Raw Tweets are collected via Twitter Streaming API with pre-defined keywords as filters. As data streams in, three modules processes each Tweet separately: (1) Topic detection; (2) Sentiment classification; and (3) User cluster inference.

Topic detection Documentation of Twitter streaming API says it will return all matched Tweets of specified keywords if the volume is less than 1% of the entire Twitter stream. However, in practice, the problem is that it sometimes returns a Tweet with none of the keywords matched. Thus, we have the module called topic detection to determine which topic the Tweets are about. Current implementation tries to use keyword match to filter out irrelevant ones. Also, it filters the Tweets by some other criteria, such as the language of the Tweets and the author, possibly geo-location of the Tweets if available,

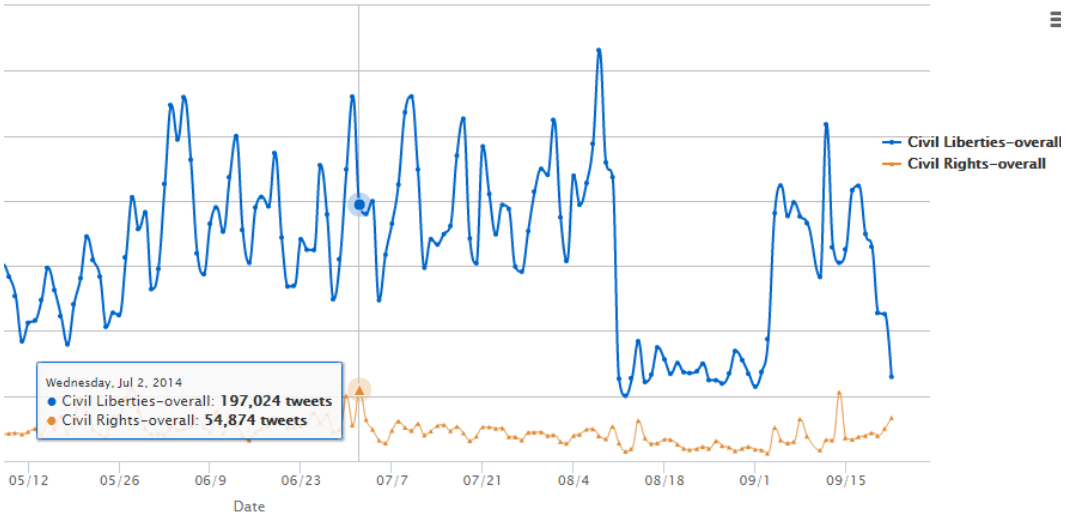

Figure 7-6: Sandbox: overlaying trends from different topics, subtopics, and sentiment on the same chart

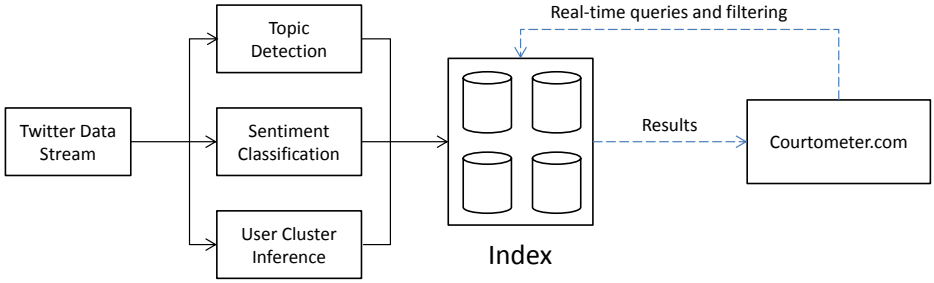

Figure 7-7: Modules and Backend Structure of *courtometer.com*.

etc.

Sentiment classification The website provides 8 types of trends to visualize: overall, positive, negative, intense, happy, angry, support, oppose. "Overall" is the count of all Tweets belonging to the topic. "Positive" and "negative" are count of Tweets with sentiment score above or below the threshold. Sentiment score is computed by taking the sum of sentiment score of each term in the Tweet. And sentiment score of each term is obtained from SentiWordNet[1]. The other 5 sentiment trends are estimated by real classifiers. Naive bayes classifier is employed here mainly because it consistently

[1]http://sentiwordnet.isti.cnr.it/

delivers the most robust results among all classifiers we tried, including support vector machine, maximum entropy and decision tree. See Chapter sentiment dynamics for more details about labeling and performance of the classifiers.

User Clustering with Inferred Characteristics Inferring characteristics for new users via LDA may require many iterations through the entire collection of profiles, even for some online algorithms [18]. However, in our setting, iterations mean either we wait for a long time before indexing the document, or the index has to be updated frequently, where either of these could greatly downgrade the efficiency. To efficiently estimate user characteristics and cluster them accordingly, we design a new simplified inference algorithm.

During the inference, Gibbs sampling gives the probability of a word w assigned to a characteristic z as follows:

$$P(z_{uw} = z|\boldsymbol{W}, \boldsymbol{Z_{-uw}}, \alpha, \beta) \propto (nw_{u,\cdot}^{z,-(u,w)} + \alpha_z) \times \frac{nw_{\cdot,w}^{z,-(u,w)} + \beta_w}{\sum_{v \in V} nw_{\cdot,v}^{z,-(u,w)} + \beta_v} \quad (7.1)$$

and the characteristic distribution of user u, θ_u will be:

$$\theta_u^z = \frac{nw_z^u + \alpha_z}{nw_\cdot^u + \alpha}$$

As explained in [36], the first component in Formula 7.1 expresses the probability of characteristic z in user u's profile, and the second component expresses the likelihood of w under characteristic z. Before we assign characteristics to any words in a user's profile, the first component cannot be estimated. Thus, we only rely on the second component to estimate proper characteristics.

Thus, the most likely characteristic $\hat{z}$ (with the highest probability) of the user u will be:

$$\hat{z} = \max_{z} \sum_{w \in u} \frac{nw_{\cdot,w}^{z,-(u,w)} + \beta_w}{\sum_{v \in V} nw_{\cdot,v}^{z,-(u,w)} + \beta_v}$$

After inferring the most likely characteristic for each user, we group users into clusters where each cluster contains users with a certain characteristic as their most likely one. In practice, we train characteristics from millions of sampled background users, and use the word-characteristic distribution to perform inference and clustering. We manually label several representative and interesting clusters to support customizing subtopics with meaningful characteristics.

7.4 Summary

The increasing prevalence of microblogs in political discourse has created a powerful new method for studying popular politics. This is particularly promising in the context of subjects, like the US Supreme Court, which do not benefit from frequent public opinion polling (as does, for example, presidential approval) and which are marked by unpredictable actions and policy roles. The infrastructure and analytic tools on *courtometer.com* establish an infrastructure for tracking discourse about the US Supreme Court (and, later, other national high courts) and measuring the public attention being paid to the Court and the sentiment the public expresses about the Court and the policy issues it decides. While these contributions have direct implications for significant lines of research in political science and judicial politics, the website's broader impacts are considerable. Further, the underlying data collection and storage infrastructure, and the analytic tools we establish can be extended to studying a variety of politically-relevant topics, such as political stability. Thus, we anticipate creating immediately useful data for political scientists studying law and courts and useful analytical tools in a variety of contexts for studying myriad problems.

Chapter 8

Conclusions and Future Work

The dynamic nature and emergence of user generated content brings great challenges to traditional models designed for static corpus, and meanwhile provides computer science and social science with unique opportunity to study and learn the patterns of evolving knowledge, trending topics, and changing sentiment. On one hand, understanding how the elements (words, topics, sentiment) change over time will gain us valuable knowledge and insights about the growth of UGC. On the other hand, information and patterns contained in the temporal dynamics could improve and inspire other applications. This dissertation presents a combination of models that learn from history and applications powered by such models.

This dissertation shows observations about how temporal information could be valuable in general. Empirical results show that (1) the words in early revisions of versioned documents are more important and tend to have increased frequencies over time. (2) Topics in Tweet streams often transit from one to another rather than stay stationary. (3) Users with similar profiles tend to have similar opinions towards political issues. (4) The early established links carry more information about retweeting behavior. All these observations and findings lay the foundation of the techniques developed in this dissertation.

Particularly, this dissertation focuses on key elements in UGC, namely word, topic, sentiment, and temporal-dependent user behavior, and develops models to describe word frequency change, topic transition pattern, and sentiment dynamics, users' temporal actions respectively. The outcomes of the developed models include improved performance of ranking versioned documents, predictive power of future topics in personal Tweet streams, characteristic-level representation of sentiment change, and better signals of users' information preference. Moreover, we build *courtometer.com* to actually apply the proposed techniques and visualize dynamics of political issues in microblogging content.

8.1 Contributions of Models

This dissertation models word, topic, and sentiment dynamics in UGC and contributes to the state-of-the-art methods in four ways:

- In previous work, words are often weighted based on their frequency in the document. Such word weighting techniques are widely used in document ranking, clustering, classification etc. We introduced Revision History Analysis (RHA) that leverages edit history of versioned documents to weight words. RHA directly captures the document authoring process when available and uses word frequency history to redefine word importance to the documents. This technique is particularly valuable for Wikipedia documents where the full revision history is accessible. RHA can be naturally incorporated into state of the art retrieval models by replacing traditional word weighting techniques with the newly defined one. Experiments show that ranking functions with RHA show consistent and significant improvements over baselines.

- Previous dynamic topic models focus on describing topic popularity change and often reinforce smoothness in the topic trends. We presented TM-LDA to explic-

itly model the topic transitions in streams of social text such as a Twitter stream for an author. The assumption of TM-LDA is that topics in personal streams transit from one to another over time, and the transitions follow certain underlying probabilities. We have shown that our method is able to more faithfully model the word distribution of a large collection of real Twitter messages and predict the future topics in individual Twitter streams compared to previous state-of-the-art methods.

- Traditional sentiment analysis methods in social media model sentiment dynamics on an individual level. We adapted topic modeling techniques to infer latent users characteristics with their profiles and observe dynamics on a characteristic-level. By leveraging social network information, we show that latent user characteristics identified by our unsupervised learning model contain more homogeneous opinions and sentiment than the ones defined by demographic attributes which often requires heavy labeling efforts.

- We analyzed the relationships between time and two major types of behavior in Twitter: posting and following. For posting behavior, we devloped UserTime model to make use of timestamps of users' posts and improve the quality of user characteristic inference over static LDA. For the following behavior, we show that early established links provide better signal to estimate retweeting preference than overall following probabilities. This finding suggests that users build up their social network by creating important links early, which could in turn help weigh links with their temporal information.

8.2 Contributions of Applications

Besides the models we developed to describe temporal dynamics in UGC, this dissertation also presents applications powered by the proposed techniques. We built *cour-*

tometer.com to visualize topic and sentiment change of popular political issues. The website provides users, especially social science researchers, with a portal to access and analyze trends of topics and sentiment. Furthermore, with the function of adding customized filters, such as additional keywords in the Tweets, locations of the authors, and the characteristics of the authors, *courtometer.com* offers a valuable tool to "zoom" into fine grained dynamics. A top Tweet function let users click on a single day and inspect the potential cause or leading opinions in the trends or sentiment. Together, our effort of translating techniques developed in this dissertation into a high-performance publicly accessible web service leads to a new way of studying public opinion and a powerful analytical tool for social scientists.

8.3 Limitations and Challenges

Dynamics in user generated content are often the results of many complex factors, including external (outside of UGC) events, user's real life experience, etc. The models developed in this dissertation have their limitations and face great challenges brought by the complexity of the problems.

- RHA model uses word frequency history to define better term weights for ranking functions. However, predicting how word frequency would change remains a challenge. Our analysis shows that frequent words in early revisions have high likelihood to appear more often after revision, but the actual prediction gains relatively low accuracy. The proposed solution is to incorporate more evidence from external sources, such as news articles and relevant webpages and build a more robust predictive model for word frequency change.

- TM-LDA model learns from historical topic transition patterns and predict future topics according to the current post from a user. However, our model is unable to

capture and predict the occurrence of new topics. Indeed, capturing new topics is a challenge in many systems and applications, especially in microblogging content. We propose to address this issue by integrating burst detection algorithms into the model so that it retrains the topic model once some unusual bursts in words are detected.

- Inferring user characteristics enables observing sentiment dynamics in meaningful user groups rather than individual level or mixed from all users. Predicting users' reactions to new events is still a difficult task. The main challenge is to collect enough historical events and the corresponding user reactions so that the model can learn from history and anticipate reactions by different user groups.

These limitations and challenges inspire the ways to improve proposed models. In the following section, we will discuss directions of future research on modeling temporal dynamics.

8.4 Future Research

Dynamics in UGC contains valuable information about evolutions of knowledge and emergence of topics. Our work and findings suggest interesting directions of future research.

Better Modeling of Dynamics in UGC Our work models dynamics of words, topics, and sentiment individually, and the natural extension in this direction is to integrate dynamics from these sources and jointly model them together. For example, a burst of certain word may imply the occurrence of a new topic or the rise of an existing one. The word may also have sentiment leanings which could result in changes of sentiment. By modeling these three elements together, we could enable the inference of dynamics from one source to another, which in turn allows us to more robustly capture the dynamics and

join the change in multiple dimensions (words, topics, and sentiment) together. Also, we can also learn the dependency between different sources. For instance, what types of words or topics tend to cause or associate with sentiment change. By studying the dependency, we could anticipate dynamics by observing the current trends.

Enhance Predictive Power through Dynamics Our models learn from historical dynamics and apply the patterns to the current context and predict the future. We believe that the predictive power is the foundation in next generation of intelligent systems, including search, recommendation, personal aide, and analysis.

- As a personalized systems, it must understand the current context and anticipate users' needs according to his historical actions. We can extend our TM-LDA model to search queries and other behavioral content on desktop and mobile to achieve this goal.

- As an analytical tool, the system should learn how different types of users react to topics and events from history, and predict user's reactions when the new topic or event occurs. Our user characteristic inference technique provides the basics for understanding user properties. By leveraging event extraction algorithms, we can link the context of events and the properties of event participants. The results from our work can help build a conditional model which takes user's characteristics and event content as input, and output the anticipated reactions.

Towards Interpreting Dynamics in UGC As we discussed, real-world events are often the driven power or the trigger for dynamics in UGC. To truly understand the resulting dynamics, the content of the events are required. A more generic event detection and extraction algorithm is needed, which can not only identify events, but understand the context of events. With identified events and captured dynamics, it is preferable to create a mapping between them to help interpret the reason of occurred dynamics.

This requires the model to understand the context of both events and dynamics, and be flexible enough to match the unaligned pairs.

8.5 Overall Summary

In this dissertation, we model the dynamics of words, topics, sentiment, and temporal-dependent user behavior in UGC. For word dynamics, we propose RHA model to leverage document edit history and redefine the word weights in ranking functions. For topic dynamics, we design TM-LDA to learn topic transitions from historical Tweet streams and apply the transition patterns to future topic prediction. For sentiment dynamics, we develop user characteristic inference models that find homogeneous opinion groups and display the opinion and sentiment change on a group-level. For user behavior in UGC, we show that the temporal information of users' posting and following actions contains valuable information about users' characteristics and information preference. We also build a public accessible website to run our proposed algorithms in real-time and visualize the dynamics in UGC. Together, this dissertation provides the fundamental techniques to model and understand dynamics in UGC and the public available analytical tool to study dynamics for social scientists and the research community.

Bibliography

[1] Eytan Adar, Jaime Teevan, Susan T. Dumais, and Jonathan L. Elsas. The web changes everything: Understanding the dynamics of web content. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 282–291, New York, NY, USA, 2009. ACM.

[2] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[3] Ablimit Aji, Yu Wang, Eugene Agichtein, and Evgeniy Gabrilovich. Using the past to score the present: Extending term weighting models through revision history analysis. In *CIKM*, pages 629–638, 2010.

[4] James Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[5] Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 3–12, Washington, DC, USA, 2008. IEEE Computer Society.

[6] Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. In *Web Intelligence*, 2010.

[7] Sitaram Asur, Bernardo A. Huberman, Gábor Szabó, and Chunyan Wang. Trends in social media: Persistence and decay. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[8] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[9] Charley Beller, Rebecca Knowles, Craig Harman, Shane Bergsma, Margaret Mitchell, and Benjamin Van Durme. I'm a belieber: Social roles via self-identification and conceptual attributes. In *Association for Computational Linguistics*, 2014.

[10] Michael Bendersky and W. Bruce Croft. Discovering key concepts in verbose queries. In *SIGIR*, pages 491–498, 2008.

[11] Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson, and David Yarowsky. Broadly improving user classification via communication-based name and location clustering on twitter. In *HLT-NAACL*, pages 1010–1019, 2013.

[12] Shane Bergsma and Benjamin Van Durme. Using conceptual class attributes to characterize social media users. In *ACL*, pages 710–720, 2013.

[13] Adam Bermingham and Alan F. Smeaton. Classifying sentiment in microblogs: Is brevity an advantage? In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1833–1836, New York, NY, USA, 2010. ACM.

[14] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 113–120, New York, NY, USA, 2006. ACM.

[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[16] John D. Burger, John Henderson, George Kim, and Guido Zarrella. Discriminating gender on twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1301–1309, 2011.

[17] Luciana S. Buriol, Carlos Castillo, Debora Donato, Stefano Leonardi, and Stefano Millozzi. Temporal analysis of the wikigraph. In *WI*, pages 45–51, 2006.

[18] Kevin R. Canini, Lei Shi, Helen Wills Neuroscience, and Thomas L. Griffiths. Online inference of topics with latent dirichlet allocation. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2009.

[19] G. Cao, J.Y. Nie, and J. Bai. Integrating word relationships into language models. In *Proc. of SIGIR*, 2005.

[20] Paula Carvalho, Luís Sarmento, Mário J. Silva, and Eugénio de Oliveira. Clues for detecting irony in user-generated contents: Oh...!! it's "so easy" ;-). In *Proceedings of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, TSA '09, pages 53–56, New York, NY, USA, 2009. ACM.

[21] Raviv Cohen and Derek Ruths. Classifying political orientation on twitter: It's not easy! In *ICWSM*, 2013.

[22] Robert Dahl. Decision-making in a democracy: The supreme court as national policy-maker. In *Journal of Public Law*, 1957.

[23] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 241–249, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[24] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC*, 2006.

[25] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41:391–407, 1990.

[26] Miles Efron. Linear time series models for term weighting in information retrieval. *Journal of the American Society for Information Science and Technology (JASIST)*, 2010.

[27] J.L. Elsas and S.T. Dumais. Leveraging temporal dynamics of document content in relevance ranking. In *Proc. of WSDM*, 2010.

[28] Robert S Erikson, John P McIver, and Gerald C Wright Jr. State political culture and public opinion. *The American Political Science Review*, pages 797–813, 1987.

[29] Santo Fortunato. Community detection in graphs. In *Physics Reports*, 2010.

[30] Charles H. Franklin and Liane C. Kosaki. Republican schoolmaster: The u.s. supreme court, public opinion, and abortion. In *The American Political Science Review*, 1989.

[31] Mihai Georgescu, Nattiya Kanhabua, Daniel Krause, Wolfgang Nejdl, and Stefan Siersdorfer. Extracting event-related information from article updates in wikipedia. In *ECIR*, pages 254–266, 2013.

[32] James L Gibson and Gregory A Caldeira. *Citizens, courts, and confirmations: Positivity theory and the judgments of the American people*. Princeton University Press, 2009.

[33] James L Gibson, Gregory A Caldeira, and Lester Kenyatta Spence. Measuring attitudes toward the united states supreme court. In *American Journal of Political Science*, 2003.

[34] AndrÍÍ Gohr, Alexander Hinneburg, Rene Schult, and Myra Spiliopoulou. Topic evolution in a stream of documents. In *SDM*, pages 859–872. SIAM, 2009.

[35] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[36] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Science*, 101:5228–5235, 2004.

[37] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum. Integrating topics and syntax. In *Advances in Neural Information Processing Systems 17*, Cambridge, MA, 2005. MIT Press.

[38] Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press, 2005.

[39] Pedro Calais Guerra, Wagner Meira Jr., and Claire Cardie. Sentiment analysis on evolving social streams: How self-report imbalances can help. In *Proceedings of the Seventh ACM International Conference on Web Search and Data Mining*, WSDM '14, New York, NY, USA, 2014. ACM.

[40] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, (1):10–18, November.

[41] Valerie Hoekstra. Public reaction to supreme court decisions. In *Cambridge University Press*, 2003.

[42] Matthew D. Hoffman, David M. Blei, and Francis R. Bach. Online learning for latent dirichlet allocation. In *NIPS*, pages 856–864, 2010.

[43] Thomas 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.

[44] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM.

[45] Jeon hyung Kang and Kristina Lerman. Using lists to measure homophily on twitter. In *AAAI workshop on Intelligent Techniques for Web Personalization and Recommendation*, July 2012.

[46] Tomoharu Iwata, Takeshi Yamada, Yasushi Sakurai, and Naonori Ueda. Online multiscale dynamic topic models. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 663–672, New York, NY, USA, 2010. ACM.

[47] Timothy R. Johnson and Andrew D. Martin. The public's conditional response to supreme court decisions. In *American Political Science Review*, 1998.

[48] Jaap Kamps, Shlomo Geva, and Andrew Trotman. Analysis of the inex 2009 ad hoc track results. In *INEX*, 2009.

[49] Shiva Prasad Kasiviswanathan, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM)*, 2011.

[50] Oren Kurland and Lillian Lee. Corpus structure, language models, and ad hoc information retrieval. In *SIGIR*, pages 194–201, 2004.

[51] John Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 111–119, New York, NY, USA, 2001. ACM.

[52] Jeffrey R Lax and Justin H Phillips. How should we estimate public opinion in the states? *American Journal of Political Science*, 53(1):107–121, 2009.

[53] Matthew Lease. An improved markov random field model for supporting verbose queries. In *SIGIR*, pages 476–483, 2009.

[54] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 631–640, 2010.

[55] Cindy Xide Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. PET: a statistical model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2010.

[56] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan Claypool Publishers, 2012.

[57] X. Liu and W. B. Croft. Cluster-based retrieval using language models. *SIGIR*, pages 186–193, 2004.

[58] Aibek Makazhanov and Davood Rafiei. Predicting political preference of twitter users. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 298–305, 2013.

[59] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schĺźtze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[60] Ramesh M. Nallapati, Susan Ditmore, John D. Lafferty, and Kin Ung. Multiscale topic tomography. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 520–529, New York, NY, USA, 2007. ACM.

[61] Le T. Nguyen, Pang Wu, William Chan, Wei Peng, and Ying Zhang. Predicting collective sentiment dynamics from time-series social media. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 6:1–6:8, New York, NY, USA, 2012. ACM.

[62] Minh-Thap Nguyen and Ee-Peng Lim. On predicting religion labels in microblogging networks. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 1211–1214, 2014.

[63] Paul Ogilvie and James P. Callan. Combining document representations for known-item search. In *SIGIR*, pages 143–150, 2003.

[64] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[65] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[66] MJ. Paul and M. Dredze. You are what you tweet: Analyzing Twitter for public health. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[67] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, republicans and starbucks afficionados: User classification in twitter. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 430–438, 2011.

[68] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, republicans and starbucks afficionados: User classification in twitter. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 430–438, New York, NY, USA, 2011. ACM.

[69] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 91–100, 2008.

[70] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. Classifying latent user attributes in twitter. In *Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents*, SMUC '10, pages 37–44, 2010.

[71] Antonio Reyes, Paolo Rosso, and Tony Veale. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1):239–268, 2013.

[72] S. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proc. 3rd TREC*, pages 109–126, 1994.

[73] Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 extension to multiple weighted fields. In *CIKM*, pages 42–49, 2004.

[74] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic 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.

[75] Ankan Saha and Vikas Sindhwani. Learning evolving and emerging topics in social media: A dynamic nmf approach with temporal regularization. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 693–702, New York, NY, USA, 2012. ACM.

[76] Svitlana, Volkova, Glen Coppersmith, and Benjamin Van Durme. Inferring user political preferences from streaming communications. In *Association for Computational Linguistics (ACL)*, 2014.

[77] Gabor Szabo and Bernardo A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53:80–88, August 2010.

[78] Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, pages 1397–1405, New York, NY, USA, 2011. ACM.

[79] Christopher Thomas and Amit P. Sheth. Semantic convergence of wikipedia articles. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 600–606, Washington, DC, USA, 2007. IEEE Computer Society.

[80] RichardJ. Timpone. Ties that bind: Measurement, demographics, and social connectedness. *Political Behavior*, 20(1):53–77, 1998.

[81] A. Trotman. Choosing document structure weights. *Information Processing and Management*, 41(2), 2005.

[82] Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, 2006.

[83] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, 2009.

[84] Chong Wang, David M. Blei, and David Heckerman. Continuous time dynamic topic models. In *UAI*, pages 579–586, 2008.

[85] M. Wang and L. Si. Discriminative probabilistic models for passage based retrieval. In *Proc. of SIGIR*, 2008.

[86] Xuerui Wang and Andrew 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)*, 2006.

[87] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. *SIGIR*, pages 178–185, 2006.

[88] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[89] Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Just how mad are you? finding strong and weak opinion clauses. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 761–767. AAAI Press, 2004.

[90] Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In *ICWSM*, 2012.