University of California
Santa Barbara

Opinion Detection, Sentiment Analysis and User Attribute Detection from Online Text Data

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Kasturi Bhattacharjee

Committee in charge:

Professor Linda Petzold, Chair
Professor Noah Friedkin
Professor Xifeng Yan

September 2016

The Dissertation of Kasturi Bhattacharjee is approved.

Professor Noah Friedkin

Professor Xifeng Yan

Professor Linda Petzold, Committee Chair

July 2016

Opinion Detection, Sentiment Analysis and User Attribute Detection from Online Text Data

by

Kasturi Bhattacharjee

To Ma, Baba - thank you for everything.

Acknowledgements

My Ph.D. has been an incredible journey that has not only taught me to be a researcher, but also helped me grow as a person. I shall attempt to thank the people who have been instrumental in making this journey a fruitful one. Words fall short to thank the two most important people in my life - my parents. I thank them for providing me with a wonderful childhood and instilling the values that make me the person I am today. Dr. Kaushik Kr. Bhattacharjee and Dr. Sumita Bhattacharjee - I am extremely proud of being your daughter and thank you wholeheartedly for your constant love and support. Miss you, Baba.

With heartfelt gratitude, I thank my advisor and constant source of inspiration, Professor Linda Petzold, for her guidance and advice throughout the course of my Ph.D. She has helped me learn how to conduct research and find my own way in a field that I am very passionate about. The zeal and energy that she brings into research is something I hope to imbibe in my own life. This Ph.D. would not have been possible without her.

During the course of my Ph.D., I have had the opportunity of working with some amazing researchers, that have led to fruitful collaborations. Thanks to Dr. I.R. Stanoi and Dr. Prithivraj Sen at IBM Research Almaden, for mentoring me during my internship and for the continued collaboration. I have learnt a lot from them. Further, I would like to thank Dr. Janet Keel and Dr. Ramasubbu Venkatesh at Target Data Science and Engineering for their incredible mentorship during my internship. I thank them for allowing me the creative freedom to solve an interesting research problem, and for their insightful suggestions that enriched the process. Further, my thanks goes to Professors Xifeng Yan and Noah Friedkin for consenting to be on my committee, and for their advice and suggestions throughout the course of my Ph.D. which helped shape and better my work. I must mention my fellow grad student and friend (and now an alumni of UCSB), Dr. Saiph Savage, who I had the pleasure of collaborating with as well.

I must also mention the wonderful people I worked with during my Teaching Assistantships

here at UCSB. Working with Dr. Diana Franklin, Dr. Matthew Buoni and Professor Omer Egecioglu was a lot of fun and I thank them for providing me the opportunity to do so. A special thanks to all my students who I enjoyed interacting with and learned a lot from. They taught me how rewarding teaching can be.

I have had my share of personal struggles throughout the course of my Ph.D., and apart from my family and loved ones, the outstanding staff of our Department have stood by me in my toughest times for which I am immensely grateful. Greta Carl-Halle, Jillian Title, Sheryl Reimers, Benji Dunson - I am so thankful to each and everyone of you for all the support you provided me with. I will truly miss you! I must also offer my sincere thanks to Professors Tim Sherwood and Elizabeth Belding for their kindness and advice as graduate advisors, especially in times of need. A special thanks also goes to Professor Ambuj Singh for his encouragement especially during the first few years of my Ph.D.

I have been fortunate to have made some amazing friends over the course of my stay at Santa Barbara. Aseem, Vineeth, Aurelie, Maha, Preshit, Saurabh, Ritesh, Shivira, Krithika - I'll never forget the long-winded discussions on almost every topic under the sun that we've enjoyed over the years. I'll cherish these friendships forever. My lab mates at the Petzold research group have provided a patient ear to all the ideas I bounced off of them throughout the years for which I am immensely grateful. Ben, Kevin, Michael (Trogdon), YY, Bernie, Tie - thanks for being such a patient audience and helping make my research presentations better. A very special thanks to my dearest friend, Veronika Strnadova-Neeley for always being there and being my partner-in-crime in everything we did together. I am so lucky to have found a friend like you!

My most affectionate thanks is reserved for my best friend and soulmate, Soumya. You've been my pillar of support through thick and thin, and I couldn't have done this without you. Being with you has enriched this journey and filled it with happiness and love. Thank you!

I consider myself incredibly lucky to have completed my Ph.D. at this gloriously beautiful

and prestigious university, and at a Department that not only excels in research but nurtures its students as well. Thanks to UCSB and to the entire Computer Science Department!

Curriculum Vitæ
Kasturi Bhattacharjee

Education

2016	Ph.D. in Computer Science (Expected), University of California, Santa Barbara.
2009	B.E. in Electronics and Telecommunication Engineering, Jadavpur University, India.

Research Interests
Analyzing user opinions and attributes from online text data using NLP and machine learning techniques.

Work Experience

- *Graduate Student Researcher*, advised by Prof. Linda Petzold, Dept.of Computer Science, UC Santa Barbara, Fall 2011-present.
- *Data Science Summer Intern*, Target Data Science and Engineering, June 2015 - September 2015.
- *Research Intern*, IBM Research Almaden, March 2015 - June 2015.
- *Research Mentor* for Summer Applied Biotechnology Research Experience (SABRE) Internship, Summer 2013.
- *Research Intern*, IBM Research India, June 2012 - September 2012.
- *Teaching Assistant* for Introduction to Computer Programming, Dept. of Computer Science, UC Santa Barbara, Winter 2016, Winter 2015, Spring 2013, Winter 2012.

Publications

- Kasturi Bhattacharjee, Prithviraj Sen and I.R. Stanoi, **Automatic Detection of Age and Conversational Topics of Twitter Users using Distributed Representation of Words**, *10th ACM Conference on Web Search and Data Mining (WSDM 2017)*, submitted.
- Kasturi Bhattacharjee and Linda Petzold, **What Drives Consumer Choices?Mining Aspects and Opinions on Large Scale Review Data using Distributed Representation of Words**, *in preparation*.
- Kasturi Bhattacharjee and Linda Petzold, **Detecting Opinions in a Temporally Evolving Conversation on Twitter**, *Social Informatics (pp. 82-97), Springer International Publishing 2015, pg 82-97, Volume 9471.*
- Saiph Savage, Andres M. Hernandez, Kasturi Bhattacharjee and Tobias Hollerer, **Tag Me Maybe: Perceptions of Public Targeted Sharing on Facebook**, *Hypertext: 26th ACM Conference on Hypertext and Social Media 2015.*

- Kasturi Bhattacharjee and Linda Petzold, **Probabilistic User-level Opinion Detection on Online Social Networks**, *Social Informatics (pp. 309-325), Springer International Publishing 2014, Volume 8851.*
- Kasturi Bhattacharjee, Soumyadeep Chatterjee, and Amit Konar, **A Novel Clustering Method for Gene Microarray Data Using Intra-Cluster Distance and Variance**, *IEEE International Advanced Computing Conference 2009 (IACC-09), March 2009.*
- Soumyadeep Chatterjee, Kasturi Bhattacharjee, Amit Konar, and Atulya Nagar, **A Robust Clustering Method for Gene Microarray Data Using Genetic Algorithm**, *European Modeling Symposium (EMS-08), September 2008.*
- Soumyadeep Chatterjee, Kasturi Bhattacharjee and Amit Konar, **A Simple and Robust Algorithm for Microarray Data Clustering Based on Gene Population-Variance Ratio Metric**, *Biotechnology Journal, Vol 1 Issue 4, Issue 9, 2009.*

Awards and Honors

- Broida-Hirschfelder Graduate Fellowship Award at UC Santa Barbara, October 2015.
- Nominated for Outstanding Teaching Assistant Award, Dept. of Computer Science, UC Santa Barbara, March 2015.
- Grace Hopper Scholarship Award by the Anita Borg Institute, July 2014.
- Ph.D. Progress Award by the Dept. of Computer Science, UC Santa Barbara, March 2014.
- Travel Award by the Institute of Mathematics and its Applications, March 2012.
- Citrix Go-To Fellowship at UC Santa Barbara, 2010.

Abstract

Opinion Detection, Sentiment Analysis and User Attribute Detection from Online Text Data

by

Kasturi Bhattacharjee

With the growing increase in the use of the internet in most parts of the world today, users generate significant amounts of online text on different platforms such as online social networks, product review websites, travel blogs, to name just a few. The variety of content on these platforms has made them an important resource for researchers to gauge user activity, determine their opinions and analyze their behavior, without having to perform monetarily and temporally expensive surveys. Gaining insights into user behavior enables us to better understand their likes and dislikes, which in turn is helpful for economic purposes such as marketing, advertising and recommendations. Further, owing to the fact that online social networks have recently been instrumental in socio-political revolutions such as the Arab Spring, and for awareness-generation campaigns by MoveOn.org and Avaaz.org, analysis of online data can uncover user preferences.

The overarching goal of this Ph.D. thesis is to pose some research questions and propose solutions, mostly pertaining to user opinions and attributes, keeping in mind the large quantities of noise present in online textual data. This thesis illustrates that with the extraction of informative textual features and the use of robust NLP and machine learning techniques, it is possible to perform efficient signal extraction from online text data, and use it to better understand user behavior. The first research problem addressed is that of opinion detection and sentiment analysis of users on a given topic, from their self-generated tweets. The key idea is to select relevant hashtags and n-grams using an l_1-regularized logistic regression model for opinion detection. The second research problem deals with temporal opinion detection from

tweets, i.e., detecting user opinions on a topic in which the conversation evolves over time. For instance, on the widely-discussed topic of Obamacare (the Affordable Care Act in the U.S.), various issues became the focal points of discussion among users over time, as corresponding socio-political events and occurrences took place in real-time. We propose a machine-learning model based on seminal work from the sociological literature that is based on the premise that most opinion changes occur slowly over time. Our model is able to successfully capture opinions over time using publicly available tweets, as well as to uncover the key points of discussion as time progresses. In the third research problem, we utilize distributed representation of words in a method that determines, from user reviews, aspects of products and services that users like and dislike. We harness the contextual similarity between words and effectively build meta-features that capture user sentiment at a granular level. Finally in the fourth research problem, we propose a method to detect the age of users from their publicly available tweets. Using a method based on distributed representation of words and clustering, we are able to achieve high accuracies in age detection, as well as to simultaneously discover topics of conversation in which users of different age groups engage.

Contents

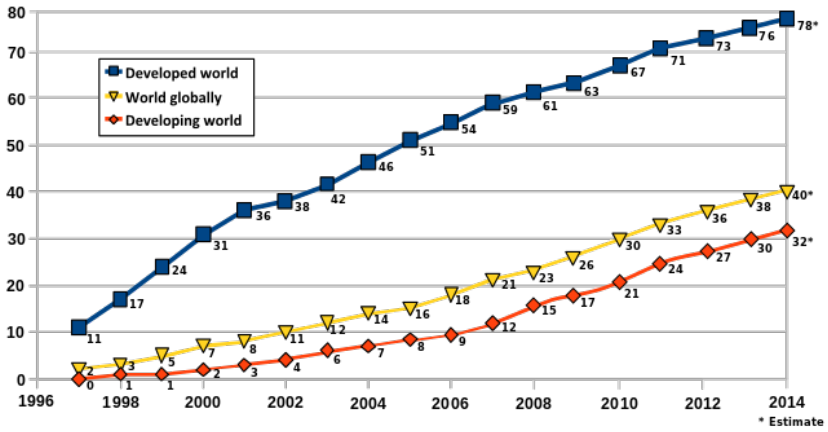
Chapter 1

Introduction

A social network is formally defined as a set of social actors, or nodes, that are connected by one or more types of relations [1, 2]. Ever since the first human societies were formed, social networks have been in existence and have played a role in influencing individual and collective behavior. However, we have been experiencing a dramatic increase in the use and popularity of online social network and other content generation platforms in the recent years. The accessibility and widespread use of the Internet in most parts of the world has lead to an enormous amount of content being generated on a daily basis by Internet users across the world. People generate content on a variety of platforms such as social networks (Facebook, Twitter, Pinterest), review websites (TripAdvisor, Yelp, Amazon), Web blogs, and countless others. Figure 1.1 shows the tremendous increase in the usage of the Internet throughout the world over the years, and Figure 1.2 illustrates the increase in popularity of social networking sites.

These online platforms are used in a myriad of different ways. For example, review websites like Amazon [4], Yelp [5] and TripAdvisor [6] are used by people to publish reviews about their purchases of products, places and services. These platforms help guide consumers in their purchase-making decisions by providing them a view of similar purchases made by other peo-

Figure 1.1: Internet Users per 100 Inhabitants Around the World. Source: International Telecommunication Union (ITU) [3].

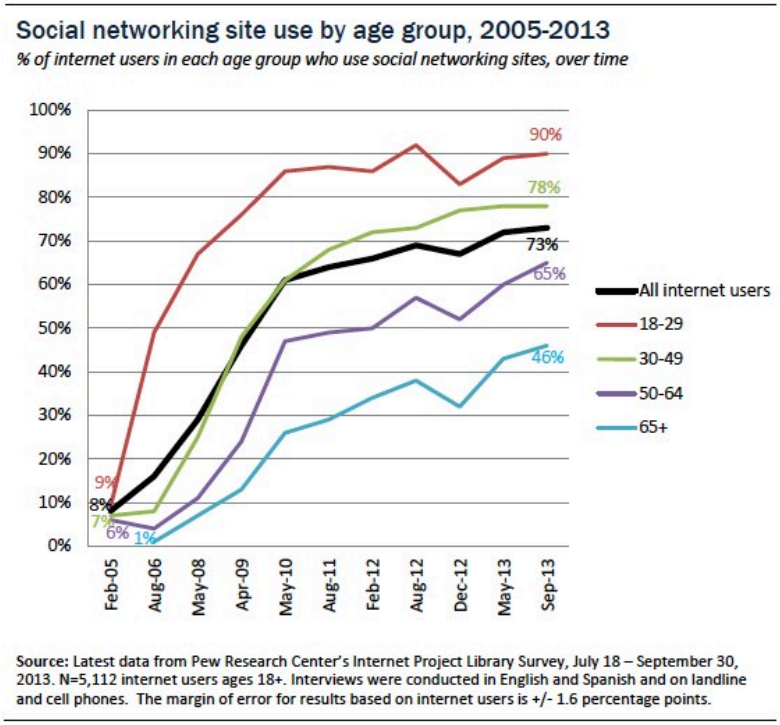


ple. On the other hand, online social networks (OSNs) are used for other purposes, ranging from communicating with friends and family, obtaining a collated view of the news of the day, advertising, etc. Further, these platforms have been recently used for socio-political reasons as well. For instance, Facebook and Twitter proved to be effective mediums of communication for protesters during the Arab Spring, enabling them to coordinate and conduct a revolution [7]. The massive popularity of social networks has led to their extensive use in political campaigns as well [8]. Further, social and political organizations such as MoveOn.org [9] and Avaaz.org [10] have emerged as platforms through which people start online petitions to increase public awareness on a myriad of important social and political issues. The variety of ways in which online platforms are used today makes them an interesting and important resource for studying user opinions and behaviour.

1.1 Importance of online user content as a research domain

This section highlights a few reasons why studying user-generated online text data has evolved into an important research area over the years.

Figure 1.2: Online social network usage over the years



- **Diversity of sources and opinions:** Different online platforms are used for various purposes, which can be studied to address different research questions. For instance, a review website such as Amazon [4] or Yelp [5] would be used for the purpose of providing reviews on purchases of products or services, while a social network such as Twitter [11] would be used for communicative purposes, and for general comments on everyday topics, including socio-political ones [7]. The knowledge that is gained from this diverse user-generated data is greatly useful from several perspectives - socio-politically, and also monetarily through viral marketing, recommendations and advertising. In addition, social networks have been shown to be fruitful in studying health and disease. For instance, Christakis and Fowler [12] have studied social factors that influence a seemingly individual trait, such as smoking.

- **Data availability:** As mentioned before, the rising popularity of online platforms has

lead to a massive amount of content generation on a daily basis. For instance, as per [13], every minute, Facebook users share about 2.5 million pieces of content, Twitter users tweet nearly 300,000 times, 200 million email messages are sent worldwide, and so on. A lot of this content is available for free for researchers to use, thereby making it a rich domain for data-driven research.

- **Network interactions:** For online social networks such as Facebook, Twitter, etc., the interactions between the users lead to interesting dynamics that has always been of great interest amongst researchers [14, 15, 16].

- **Temporal dynamics:** Since the content on these online platforms is being continuously generated over time, it allows for the unfolding and analysis of temporal processes. Thus, event detection, topic evolution and opinion dynamics can be studied using such data.

- **Need for Automation:** The enormity of the data makes it infeasible for signal extraction to be a manual process, hence it needs to be automated. Thus, there is a need for the development of methods to automatically extract information from online textual data.

1.2 Sources of Data

When conducting research on online social networks, it is important to be aware of the most popularly used platforms since they would act as the richest sources of information, but ease of data access also needs to be borne in mind. Here, I shall briefly discuss the most commonly used platforms for user-generated data, and the data accessibility issues associated with each.

As per the statistics released by statista.com [17], as of April 2016, Facebook is the largest online social network platform with over 1.5 billion active users. Although research has been conducted in the past using Facebook data [18, 19, 20], in general, owing to strict user privacy settings, data access is very restricted.

Twitter, a popular microblogging tool, can be considered by far the most studied OSN [21]. The existence of a well-defined public interface for software developers to extract data from the network [22], the simplicity of its protocol, and the public nature of most of its content are some of the reasons why it has been widely studied. However, since the beginning of the service, rate policies have been created to control the amount of data allowed to be collected by researchers and analysts. This has had a direct impact on research. While initial studies had access to all the content published in the network, more recent studies are usually limited by those policies [21].

It is also worth mentioning the existence of Chinese counterpart services for Facebook and Twitter, like Sina-Weibo [23], the largest one, with more than 500 million registered users [24]. Although the usage of those services may differ due to cultural aspects [25, 26], similar lines of inquiry can be developed in both the western and eastern equivalents [27, 28, 29].

Review websites such as Amazon [4], Yelp [5], etc. host user reviews on products and services and are rich sources of online content. Yelp [5] releases a portion of its data every year for an academic challenge [30], containing user reviews on a number of businesses. Similarly, portions of the Amazon [4] reviews are available from various academic research groups [31].

Other web services that integrate social networking features have been the focus of studies. Examples are media sites like YouTube10 [32] and Flickr11 [33, 34], and news services such as Digg12 [35, 36]. Research was also conducted over implicit social networks derived from the Enron email dataset [37], university pages [38, 39] or blogs [40], even before the creation of social networking services.

1.3 Social Network Research

Online text data provides a rich source of information for gaining insights into user opinions and behavior. It provides us with a variety of content which would otherwise have to be

obtained by conducting surveys that can be expensive and time-consuming. Further, online data provides us access to a large number of users and their self-generated content. Thus, it is a rich resource for analysing user behavior.

Chapter 2 of this thesis contains a detailed overview of some of the primary areas of research in this area. The research may be primarily divided into two groups: Social Data Analysis, in which the textual content is the focus of the study, and Social Interaction Analysis, in which the user network is the primary object of study. Our research focuses on Social Data Analysis, with emphasis on user opinion mining and sentiment analysis, and user profiling. In particular, we conducted sentiment analysis on various datasets such as Twitter posts, Yelp and Amazon reviews, and on a number of topics such as U.S. Politics, the Affordable Care Act, Immigration Reforms in the U.S. on the Twitter front and restaurant and camera reviews from Yelp and Amazon. Further, we analyze Twitter data to detect user ages by identifying topics of conversation they engage in.

1.4 Challenges and Unresolved Areas

In spite of the large body of work in this domain, many of the primary challenges in working with online textual data have not been fully resolved. Here, we discuss a few of those challenges.

- **Noise:** User-generated online text data is inherently noisy in nature [41, 42]. Since people tend to use informal language when expressing themselves online, noise easily creeps in and can make information extraction very difficult. Noise includes the presence of misspellings, acronyms, slangs, case insensitivity, misplaced or multiple punctuation marks, etc. to name a few. A few examples shall illustrate the point:
  - **Twitter:** “Sometimes I snap at *ppl* on twitter *bcuz im* insecure. *its* a defense *mag-*

nesium."

- **Twitter:** "I wanna visit the *ifold* tower in france one day."
- **Facebook:** "*Illiturate* people are making me sad. How hard is it to actually spell the words correctly*???*"

- **Ambiguity:** When online surveys are conducted, participants are usually presented with a set of questions to answer. This results in a very structured format and makes signal extraction easy. However, online platforms do not possess such a structure, which often results in the generation of ambiguous content from which it becomes hard to decipher meaning. The following is an example of a camera review on Amazon:

  Review: *"I recieved this camera a couple of days ago and I am very imprissed at the ease of it and the sftware. The quality of the pics are really stunning...I am looking like a professional photographer alreaddy, and this is the first time I have picked up a real camera since high school photography class...25 years ago...sigh; I have really checked out the prices online and off, and this is a great buy! I dont know how you do it Amazon. Keep bringing us the BARGAINS! Well, I am off to buy a compact flash memory card...ta ta!"* **Rating:** 3.0

  On reading the review, it would seem like the user was very impressed with her purchase, but the rating accompanying it does not reflect that.

- **Sparsity:** Owing to the informal nature of the language used in online text data, there are are many ways of expressing the same concept. For instance, the word *like* may be expressed in several other ways such as *enjoy*, *admire*, *love*, etc. In this case, considering each word as a feature leads to high-dimensional, sparse data matrices.

- **Difficulty in signal extraction:** OSNs are platforms where users can express their generic likes and dislikes on a myriad of topics. In such a scenario, it is challenging

to extract information pertaining to the research topic in question.

- **Absence of large-scale annotated datasets:** There is a lack of large-scale annotated datasets in this area, which makes it hard to obtain ground truth with which to train and verify models. This is a major problem in conducting research in this area.

- **Time:** There is an inherent temporal aspect to the data in this area, since online posts continue to be generated everyday. Not many methods exist to capture and analyze this temporal nature of conversation.

- **Generalizability:** One of the primary problems in this area was the discovery that most methods that analyze user behavior and sentiments are not generalisable. From our own experience, sentiment analysis methods that were previously developed did not perform as expected when applied to our problem (Chapter 3.5.2). We have attempted to address this issue by validating our proposed method on multiple datasets, for specific classes of problems.

1.5 Scope and Outline of this Thesis

This thesis provides a general overview of the research questions that have been explored in this domain in Chapter 2, and then delves into the details of our own research projects. Chapter 3 addresses the problem of opinion detection from Twitter data on specific topics, and then proposes a solution. It covers in detail the steps undertaken for the process, starting from the data collection up until the model was formulated and verified. In Chapter 4, we address the concept of time with respect to online conversations on a topic and propose a solution on detecting opinions in the face of an evolving conversation. The method proposed has its roots in seminal sociological models of the past. For both Chapters 3 and 4, Twitter [11] data is used for conducting the experiments. Further, in Chapter 5, we explore the use of a review dataset for

performing Aspect-Based Sentiment Analysis. The dataset, obtained from Yelp [5], contains user reviews on businesses and services. This work entails understanding user opinions but with a deeper granularity. We propose a method to detect what users like and dislike about the item under review and why, using distributed representation of words. Additional experiments using the same method outlined in Chapter 5 are conducted on a digital camera review dataset from Amazon, the results of which are presented in Appendix A. In Chapter 6, we address the problem of identifying user age by analyzing their tweets. Using publicly available user tweets, we propose a method that allows us to discover the various topics of conversation that users of different age groups engage in, which in turn allows us to detect their age. Finally, Chapter 7 provides a conclusion.

1.6 Permissions and Attributions

1. The work of Chapters 3 and 4 were performed in collaboration with my advisor, Prof. Linda Petzold. These have been published in [43] and [44] respectively.

2. The content of Chapters 5 and Appendix A are a result of a collaboration with Janet Keel and Ramasubbu Venkatesh of Target Data Science and Engineering, along with my advisor, Prof. Linda Petzold. This work is in preparation for a publication.

3. The research for Chapter 6 was performed in collaboration with I. R. Stanoi and Prithviraj Sen of IBM Research Almaden. This work has been submitted to the 10th ACM International Conference on Web Search and Data Mining 2017 [45].

Chapter 2

Survey and Related Work

This Chapter presents an overview of the existing work in this domain, from a computational point of view. The research in this domain may be broadly categorised into two areas: research that focuses on the textual content rather than the network, and research that is more geared towards the network and interactions amongst users. The following sections will present an overview of each of these areas and the research questions.

2.1 Social Data Analysis

Social Data Analysis focuses on the content that is generated by users. Working with noisy online text requires extensive use of NLP techniques and machine learning. In addition to the challenge of building robust methods to solve the problems, scalability is a big issue owing to the large amounts of social data that is available at the disposal of researchers.

2.1.1 Opinion Mining/Sentiment Analysis

Opinion mining and sentiment analysis focus on understanding user opinions and sentiments from online posts. This has been one of the most active areas of research in this domain.

Owing to the growing popularity of online social networks, review sites and personal blogs, there are now a variety of sources from which to obtain user-generated data, each of which is used for different purposes as well. Researchers have attempted to estimate user opinions on products they have purchased [46, 47, 48, 49], on political opinions expressed on online social networks (OSNs) [50], or general positivity or negativity conveyed through user posts [51]. Two survey papers that provide extensive coverage on this area are by Pang and Lee [52] and Liu and Zhang [53].

Research involving sentiment analysis or opinion mining on social networks may be divided into two areas: techniques that are based on lexicons of words, and techniques that are based on machine learning. The lexicon-based methods work by using a predefined collection (lexicon) of words, where each word is annotated with a sentiment. Various publicly available lexicons are used for this purpose, each differing according to the context in which they were constructed. Examples include the Linguistic Inquiry and Word Count (LIWC) lexicon [54, 55], the Multiple Perspective Question Answering (MPQA) lexicon [56, 57, 58], SentiStrength [59] and SentiWordNet [60, 61]. These lexicons typically contain sentiment annotated words, usually in the form of a numeric score. Given some text, a "sentiment score" can be computed using the words occurring in the text and their respective scores in the lexicon concerned. This has been used to perform sentiment analysis for social network data, movie reviews, blog posts, etc. [62, 63, 64, 65].

Machine learning based sentiment analysis typically includes the use of supervised approaches which involve feature extraction, followed by the formulation of a classification problem where the labels of the classifier refer to the sentiment expressed by a user on a particular topic. Features extracted may include bag-of-words or n-grams [66, 67], Parts-of-Speech tags [68, 69], dependency-tree-based features [70] etc. Commonly used methods such as Maximum Entropy, Naive Bayes, SVM, etc. [51] are used as classification methods.

Sentiment analysis on product review data, usually referred to as Aspect-Based Sentiment

Analysis, is another area of research and typically involves uncovering what users like/dislike about the products they have reviewed. Such efforts adopt either a topic-modeling based approach [46, 47, 71, 72], or use feature-extraction, such as Parts-of-Speech Tagging [48, 49, 73] to identify nouns as "aspects" of the products being reviewed. This problem is explored in this thesis, and details will be provided in Chapter 5.

2.1.2 Trending Topic Detection

Trending topics are typically driven by emerging events, breaking news and general topics that attract the attention of a large fraction of users on online social network platforms like Twitter. Cheong and Lee [74] analyze tweets to research the anatomy of trending topics. They split them into 3 categories: *long-term*, *medium-term* and *short-term* topics. Long-term topics occur infrequently, but over a long amount of time in the public time-line, while medium-term topics occur more frequently but are limited to a time range of a few days. Short-term topics are heavily discussed topics, and often refer to current events. Further, Cheong and Lee [74] categorize users into 3 major groups: "Personal" (a majority of whose postings are on personal topics), "Aggregator" (those who collect and publish information, such as news agencies, politicians etc.) and "Marketing" (those that work to promote a product but also lead to spam and unsolicited postings). The results show that it is mostly users who talk about their personal life that contribute to emerging trending topics.

In [75], the authors develop a system called TwitterMonitor to identify emerging trends on Twitter in real time. A trend is identified as a set of bursty keywords that occur frequently together in tweets. TwitterMonitor provides meaningful analytics that synthesize an accurate description of each topic. It extracts additional information from the tweets that belong to the trend, aiming to discover interesting aspects of it, such as tracking the popularity of the trend over time and the origin of geographically focused trends. Users interact with the system by

ordering the identified trends using different criteria and submitting their own description for each trend.

Benhardus [76] outlines methodologies of detecting and identifying trending topics from streaming data. Term frequency-inverse document frequency (TF-IDF) analysis and relative normalized term frequency analysis are performed on tweets to identify the trending topics. Relative normalized term frequency analysis identifies unigrams, bigrams, and trigrams as trending topics, while term frequency-inverse document frequency analysis identifies unigrams as trending topics.

2.1.3 Event Detection

Online social network platforms act as an important channel for reporting world events, which makes them a useful resource for detecting events such as health epidemics and natural disasters. The work of Culotta [77] explores the possibility of detecting influenza outbreaks by analyzing Twitter data. The author uses a bag-of-words classifier in order to predict influenza-like illness (ILI) rates in a population, based on the frequency of messages containing certain keywords. He compares rates with the U.S. Centers for Disease Control and Prevention (CDC) statistics.

Paul and Dredze [78] propose an Ailment Topic Aspect Model (ATAM) for extracting general public health information from millions of health related tweets. The approach discovers many different ailments (diseases), such as flu and allergies, and learns symptom and treatment associations. This model discovers a larger number of more coherent ailments than Latent Dirichlet Allocation (LDA) [79]. It produces more detailed ailment information (symptoms/treatments) and tracks disease rates consistent with published government statistics (influenza surveillance) despite the lack of supervised influenza training data. Their work was further utilized in [80] for the discovery of several more ailments, including allergies, obesity

and insomnia.

On the detection of natural events, Sakaki et al. [81] investigate the real-time interaction of events such as earthquakes on Twitter, and propose an algorithm to monitor tweets and detect a target event. To detect a target event, they devise a classifier of tweets using a support vector machine [82] based on features, such as the keywords in a tweet, the number of words, and their context. Subsequently, they produce a probabilistic spatiotemporal model for the target event that can find the center and the trajectory of the event location. They can detect an earthquake with high probability (96% of earthquakes of Japan Meteorological Agency (JMA) seismic intensity scale 3 or more are detected) merely by monitoring tweets.

2.1.4 User Profiling

With the huge number of users that participate in online review sites and social networks, companies are very interested in understanding their users better to be able to tailor products and services according to their needs. Thus, personalization has become an important research area, and building user profiles is an important part of the process [83, 84]. User profiling may be defined as the process of extracting features that best represent the user, so as to capture their interests in the best possible way.

Identifying basic attributes of a user, such as age, gender, etc. from their online content is a key step in this process. In [85], the authors show a connection between the language that people use on Twitter and their age, and use that to detect their age. Researchers have also explored the online behavior of users, i.e. the pages they navigate, the amount of time spent on them, etc. [86] for the purpose of profiling. In addition, social information, such as social connections with other users or groups and pages, social behaviours like shares, clicks, and likes between users has been utilized in building user profiles [87, 88, 89].

2.2 Social Interaction Analysis

Social Interaction Analysis refers to the body of work that involves taking into account the user interactions and not merely the user content. Thus, the "network" aspect of the data is taken into consideration in this research area. The following are some of the primary research questions of this domain.

2.2.1 Social Influence

As in real life, in OSNs as well, users tend to be influenced by their "friends" and connections. Social influence refers to the behavioral change of individuals affected by others in a network. Social influence is an intuitive and well-accepted phenomenon in the study of social networks [90, 91, 92]. Social influence is what ultimately leads to the spread of ideas and opinions through a network, making this a key area of research.

Identifying influential users is one of the primary research directions in this area. Cha et al. [93] discuss three metrics aiming to quantify users influence in OSNs: number of connections (nodes degree), number of mentions, and number of messages reshared by other users. A discussion of the most appropriate ways to measure influence is made, revealing that simple metrics such as number of connections can be misleading to represent the future influence of a user. Weng et al. [94] were more optimistic, showing that an adaptation of the PageRank algorithm [95] can be used to successfully measure influence on networks. However, Bakshy et al. [96] showed that even though it is possible to identify influential users able to repeatedly start widely scattered cascades, determining a priori which users will influence a cascade process is a hard task.

2.2.2 Information Flow

This area of work is based on the notion of user influence in networks as discussed in Section 2.2.1. Since a user is influenced by her network neighbors, she is likely to adopt ideas and opinions from them, and propagate them to users whom she in turn influences. This leads to a flow of information and ideas across the network, which is the focus of the research conducted in this area.

This area of work may be classified into 3 categories: *Threshold Models*, *Information Cascade Models* and *Epidemic Models*. Granovetter and Schelling were the first to propose the Threshold models [97, 98], and several variants of these models have been explored since [14, 99, 100, 101]. The basic premise of the Threshold models is that at least a certain number of network neighbors (determined by the threshold) have to adopt an idea before a user decides to adopt it. Information cascades are one of the most studied phenomenons for OSNs. These refer to a contagious process in which users, after having contact with a content or a behavior, reproduce it and influence new users to do the same. This decentralized process often causes chain reactions with great proportions, involving many users and being one of the main strategies for information diffusion in social networks. Researchers have attempted to analyze information cascades and unearth the reasons that lead to their formation [102, 16, 15].

Epidemic models, as the name suggests, are models that were initially developed to study the spread of communicable diseases and epidemics, but have been found to be instrumental in studying information flow processes in social networks as well [103, 104, 105, 106]. These models generally assume that the population can be divided into different states depending on the stage of the disease [107, 108, 109, 110], such as Susceptibles (denoted by S, those who can contract the infection), Infectious (I, those who contracted the infection and are contagious), and Recovered (R, those who recovered from the disease). Simple models for disease epidemics have been applied to social spreading phenomena [111, 112, 103, 104, 105]. Rumor

spreading is one such area in social networks where epidemic models have been applied. According to [113], individuals can be in one of three possible states: ignorant (S, equivalent to susceptible in SIR), spreader (I, equiv- alent to infected) and stifler (R, equivalent to removed).

2.2.3 Modeling Opinion Dynamics

As has been discussed earlier, users express their opinions through their online content, and since they are influenced by their network neighbors, their opinions might be subject to change gradually over time. Models from a myriad of fields (Statistical Physics, Computer Science, Sociology etc.) have attempted to capture the phenomenon of opinion dynamics in a network [90, 114, 115, 91, 116, 117, 118]. Degroot's work [90] was the earliest in this class of models, according to which the opinion of a user is the weighted average of her friends' opinions. The Voter Model [114, 115] is another widely used opinion dynamics model which postulates that at each step, every user changes her opinion by choosing one of her neighbors at random and adopting the neighbor's opinion. Similar to Degroot's model [90], this model holds the same key property that a user is most likely to change her opinion to that which occurs most often in her neighborhood.

Another seminal work in this field is the Social Influence Network Theory [91] postulated by Prof. Noah Friedkin and Eugene Johnson. This model, similar to DeGroot's model, accounts for the effect of interpersonal influences for opinion change, but also factors in an anchorage to a user's initial opinion. This latter factor is determined by how susceptible a user is to changing her opinion. This work is discussed in more detail in Section 4.3.1, and along with Degroot's model [90], forms the basis of the method we propose in Chapter 4 and [44] to detect changing opinions on Twitter. Several other models, especially from the Statistical Physics domain, have been proposed for opinion dynamics as well [116, 117, 118].

2.2.4 Social Recommendation Systems/Crowdsourcing

Recommendation systems have been a popular and active area of research for quite some time [119, 120, 121]. Their utility lies in the fact that they help consumers in their purchase decisions by recommending them products or services catered to their needs and preferences, thereby saving them the time and effort of searching through a large database of items in online marketplaces. Social recommendations have become popular of late, owing to the growing popularity of social networks themselves. This stems from the fact that since users in the physical world are likely to seek suggestions from their friends before making a purchase decision, and user's friends consistently provide good recommendations [122], relationships in OSNs can be potentially exploited to improve the performance of online recommender systems[123, 124, 125, 126]. A good overview of existing social recommendation systems is presented in [127].

Most existing social recommender systems are based on collaborative filtering (CF) techniques, the underlying assumption of which is that if users have displayed similar preferences with each other in the past, they are more likely to agree with each other in the future than to agree with randomly chosen users. Memory-based CF techniques first obtain a set of correlated users for a given user they wish to recommend items for, and then aggregate ratings obtained from this set of users to estimate the missing ratings for that user. Different metrics such as Social based Weight Mean [128, 129], TidalTrust [123], MoleTrust [130], TrustWalker [125] are used to compute the set of correlated users from a social network.

Model-based CF methods mostly rely on matrix factorization techniques. Again, the motivation behind these methods is that users' preferences are similar to or influenced by users whom they are socially connected to. In co-factorization methods [124, 131], the underlying assumption is that the i-th user u_i should share the same user preference vector in the rating space (rating information) and the social space (social information). Social recommender sys-

tems in this group perform a co-factorization in the user-item matrix and the user-user social relation matrix by sharing the same user preference latent factor. Regularization methods focus on a users preference and force her preference vector to be closer to that of users in her social network [132, 126].

Thus we find that this domain has a plethora of research questions and directions of interest. Having provided an overview of the primary areas of research, we will now be introducing our own research, which lies primarily in the area of opinion mining and detecting user attributes.

Chapter 3

Probabilistic User-level Opinion Detection on Online Social Networks

3.1 Introduction

Understanding online user behavior entails comprehending user opinions and sentiments on various topics, from their self-generated posts. This chapter presents a method we developed to detect opinions of randomly selected sets of users on a given topic, from their publicly available tweets [43]. The topics under consideration are U.S. Politics and the Affordable Care Act (Obamacare), for which we crawled the tweets of randomly picked users over a period of time. These topics were chosen since they were widely discussed at the time of this work, and were of national importance within the US. Moreover, since both the topics affected a large section of people within the country, and usually garnered polarizing views, we expected users to vocalize their opinions strongly on social network platforms, thereby making opinion detection feasible.

As has been discussed in Section 1.4, extracting signal from online text is a non-trivial problem owing to the abundance of noise and sparsity [133, 134, 135, 42, 41]. Twitter has

gained popularity among researchers due to its emergence as one of the most widely used social networks, and also because it allows for the crawling of some of its data. However, this data also brings along with it a host of challenges. The short length of a tweet, the abundance of grammatical errors, misspelt words, informal language and abbreviations make it difficult to extract the opinion expressed through a tweet accurately. Owing to this, efficient data preprocessing and feature extraction are vital steps to the task at hand.

Further, because the opinions detected on the basis of a *single tweet* are unreliable, we focused instead on assessing the opinion of a user by aggregating the information in *all* of their tweets relating to the topic of interest over a given time period. We used a probabilistic classifier, regularized to avoid overfitting [136], to classify user opinions as positive or negative on a given topic. We found that combining the use of hashtags and n-grams was highly informative in detecting user opinions. It is to be noted here that our method requires no prior manual selection or labeling of features. On implementing our method to detect opinions on both the topics mentioned above, we obtained a high level of accuracy, which exhibits the robustness of our methodology.

3.2 Related Work

As has been briefly mentioned in Chapter 3, research involving sentiment analysis or opinion mining on social networks may be divided into two areas: techniques that are based on lexicons of words, and techniques that are based on machine learning. The publicly available lexicons include the Linguistic Inquiry and Word Count (LIWC) lexicon [55, 54] and the Multiple Perspective Question Answering (MPQA) lexicon [56, 58, 57]. The LIWC lexicon contains words that have been assigned into categories, and matches the input text with the words in each category [137]. The MPQA lexicon is a publicly-available corpus of news articles that have been manually annotated for opinions, emotions, etc. These lexicons have been widely

used for sentiment analysis across various domains, not just specifically for social networks [138, 139, 140]. Other popular sentiment lexicons that have been designed for short texts are SentiStrength [59] and SentiWordNet [61, 60]. These lexicons have been extensively used for sentiment analysis of social network data, online posts, movie reviews, etc. [63, 64, 65, 62]. However, as shown later in Section 3.5.2, they do not perform very well when applied to our problem of assessing user opinion.

Machine learning techniques for sentiment analysis include classification techniques such as Maximum Entropy, Naive Bayes, SVM [51], k-NN based strategies [141], and label propagation [142]. These usually require labeling of data for training, which is accomplished either by manually labeling posts [142], or through the use of features specific to social networks such as emoticons and hashtags [51, 141]. Some of the existing research combines lexicon-based methods and machine-learning methods [143]. These papers address a different (but related) problem than ours in that they perform tweet-level as opposed to user-level sentiment analysis. In Section 3.5.2, we compare our method to user-level sentiment generated via tweet-level sentiment obtained by the methods of [143] and [51].

The methods in [144, 145, 50] perform user-level sentiment analysis. The method in [145] uses features derived from four different types of information of a social network user: user profile, tweeting behavior, linguistic content of the messages and the user network. Our method focuses on extracting informative features from only a user's tweets, and can achieve high accuracies with a smaller number of features and a simpler model. The methods in [50] determine the political alignment of Twitter users using their tweets, as well as their retweet networks. The dataset is selected by first creating a set of politically discriminative hashtags that co-occur with the hashtags *#p2* ("Progressives on Twitter 2.0") and *#tcot* ("Top Conservatives on Twitter"). The tweets selected for the dataset carry at least one of the discriminative hashtags. In contrast, we selected our dataset via identification of users who use the generic keywords in Table 3.1 at least once, which does not require the determination of discriminative words or

hashtags. Moreover, [50] does not conduct any study on using combinations of hashtags and n-grams as features, which we have found to yield the best performance in opinion detection across two different topics (as described in Section sec: op detect expts). Thus the results are not directly comparable. In addition, our method performs automatic feature selection, which [50] does not address. In [144], user-level sentiment analysis is performed using the users' following/mention network information. Since our dataset consists of randomly chosen users, we do not have the entire neighborhood of any user.

3.3 Data Collection and Pre-processing

In this section, we discuss the method for crawling tweets on the topics of interest and the subsequent pre-processing of the data for the problem.

3.3.1 Data Collection

As mentioned before, we focused on two popular (at the time) and divisive topics for which people were more likely to voice their opinions on social media: U.S. Politics and Obamacare. For each of the topics of interest, we randomly selected users and collected their tweets over a period of time using the Twitter REST API [146]. For U.S. Politics, our tweets were collected over the period of January 2012 to January 2013, which coincided with the political campaigns leading up to the November 2012 U.S. Presidential election. For the dataset on Obamacare, we crawled tweets for 6 weeks over the months of June and July 2013.

To extract topical tweets, we filtered out tweets that contained words related to the topic of interest. For instance, for political tweets, we used words related to political figures, parties, causes or issues, or commentators whose bias is well-known. This approach is similar to that used by Romero, Meeder and Kleinberg in [147].

Table 3.1 shows the list of keywords used to obtain both the datasets and the categories

Table 3.1: Keywords used to filter out topical tweets

Dataset	Keyword	Keyword Type
U.S. Politics	*obama*	Political figure
	democrat	Political party
	p2	Political party
	romney	Political figure
	gop	Political party
	tcot	Political party
Obamacare	*obamacare*	Term for affordable health care
	koch	Industrialists who were against Obamacare
	affordable care	Term for affordable health care

that they belong to. The political dataset thus obtained was composed of 672,920 tweets from 552,524 users. The Obamacare dataset consisted of 187,141 tweets from 65,218 users.

3.3.2 Data Preprocessing

Twitter data is inherently noisy and filled with abbreviations and informal words. We clean and pre-process the dataset in the following manner to enable a better extraction of features.

- **URL removal:** In our method, URLs would not contribute to the feature extraction and were therefore removed.
- **Stopword removal:** Stopwords such as “a”, “the”, “who”, “that”, “of”, “has”, etc. were removed from the tweets before extracting n-grams, which is a common practice.
- **Punctuation marks and special character removal:** Punctuation marks such as “:”, “;” etc. and special characters such as “[]”, “,”, “”, etc. were removed before extracting n-grams.
- **Additional whitespace removal:** Multiple white spaces were replaced with a single whitespace.

- **Conversion to lowercase:** Tweets are not generally case-sensitive owing to the informal language used. For instance, users may use either "Obama" or "obama" when referring to the current U.S President Barack Obama. Thus, we converted the tweets to lowercase to preserve uniformity in feature extraction.

- **Tokenization:** The tweets were tokenized into words to extract n-grams from them.

3.4 Methodology

In this section, I shall discuss in detail the problem definition and the method we proposed to solve it.

3.4.1 Problem Definition

We adopted a probabilistic view for the user opinion in that we assumed it to be a distribution over *positive* and *negative* types. On the topic of US politics, we arbitrarily defined *positive* to mean that the user was pro-Obama or anti-Romney, and *negative* to mean that she was anti-Obama or Pro-Romney. On the topic of Obamacare, *positive* was again arbitrarily defined to be a pro-Obamacare opinion, and *negative* was defined to be an anti-Obamacare opinion.

The main challenges involved were: (1) to determine appropriate features that carry information about the user's opinion (2) to learn a model that, with a sufficiently high accuracy, predicts the probabilistic user opinion from the features.

Thus, the problem definition may be summarised as: *Given a user's tweets over time on a topic, compute probabilities of her having a positive or a negative opinion.*

3.4.2 Ground Truth Labels:

We randomly picked users from each of the datasets, and then assigned a positive or negative opinion label to them by manually reading *all* of their tweets. We labeled only those users whose opinion could be unambiguously determined from their tweets. We randomly chose 490 users (222 positive and 268 negative) for our labeled dataset on U.S. Politics, and 201 users (90 positive and 111 negative) for our labeled dataset on Obamacare.

3.4.3 Proposed Method

We cast the problem at hand as a supervised binary classification problem in which the classifier outputs the probabilities of the opinions that a user can have. Logistic regression is a well-known and widely used probabilistic machine learning tool for classification. Given a binary output variable y and a set of features X, logistic regression estimates the conditional distribution $P(y = 1|X; \beta)$, where β represents the parameters that determine the effect of the features on the output.

Logistic regression utilizes the following transfer function between X and y:

$$P(y = 1|X, \beta) = h_\beta(X) = \frac{1}{1 + \exp(-\beta^T X)}. \tag{3.1}$$

To estimate the parameter β of the logistic model, we use Maximum Likelihood Estimation. Assuming that we have *m* i.i.d training samples $(y^i, X^i), i = 1, \ldots, m$, the log likelihood is given by

$$\log P(y|X, \beta) = \sum_{i=1}^{m} (y^i \log(h_\beta(X^i) + (1 - y^i) \log(1 - h_\beta(X^i)). \tag{3.2}$$

The loss function, which is the negative log-likelihood, being convex, can be minimized to estimate the optimum β, given by $\hat{\beta}$. We add a regularization to the loss function to avoid overfitting, as discussed below.

Thus, given a set of features X and a set of known outputs y in the training data, the logistic regression model learns the parameter β that determines the relationship between X and y. Once the model has been learned, it can then be used to predict the outcomes of the test data, given their features X.

Logistic regression with l_2 regularization

To avoid overfitting [136], we added a user-specified regularization term $\lambda\|X\|_2^2$ to our loss function, where $\lambda > 0$ is the regularization parameter [136]. The loss function thus becomes:

$$L(\beta) = -\log P(y|X, \beta) + \lambda\|\beta\|_2^2. \tag{3.3}$$

Logistic regression with l_1 regularization

We also explored the use of l_1-regularization [136]. This results in the loss function:

$$L(\beta) = -\log P(y|X, \beta) + \lambda\|\beta\|_1. \tag{3.4}$$

We used the open-source machine learning tool in Python, scikit-learn [148] to implement logistic regression with l_1 and l_2 regularizations. The selection of λ is discussed in Section 3.5.

3.4.4 Features for Classification

Deriving features from the tweets is a crucial step for successfully determining a user's opinion. Hashtags have become a very popular feature in Twitter and other social media sites. A hashtag is essentially a word that is prefixed with a # symbol that can be generated by a user and used in their tweets. #*followfriday*, #*mtvstars*, #*ipad,* #glee are examples of some popular hashtags on Twitter. The concept of hashtags was introduced in order to index tweets of a similar topic together, to make it easier for users to start a conversation with each other.

Apart from highlighting the topic of a tweet, hashtags have been found to carry some additional information regarding the bias of the tweet itself [141, 142]. For example, hashtags such as *#ISupportStaceyDash*, *#iloveapple*, *#twilightsucks* all carry information about the topic of the tweet and also clearly exhibit the bias of the user. A manual inspection of our dataset suggested that hashtags might be used to provide information about the bias of the tweet. For example, hashtags such as *#romneyshambles*, *#gopfail*, *#defundobamacare* were more likely to occur in tweets in which the user portrayed a negative opinion towards the respective topic. Similarly, hashtags such as *#iloveobama*, *#istandwithobama*, *#getcovered* occurred most often with tweets that carried a positive opinion towards the respective topic. For this reason, our first choice for features to use was hashtags.

Although hashtags are powerful carriers of sentiment information, sometimes they may not be sufficient to convey the bias hidden in the tweet. For instance, hashtags may just refer to a political party without seemingly carrying any bias, in which case the information we seek may be carried by the text of the tweet. Here is an example of such a tweet:

"*@MittRomney's refusal to release details of, well, anything, prove his cowardice & unfitness for the presidency.* ***#connecttheleft #gop***"

In the above tweet, the hashtags used are *#gop* ("Grand Old Party") and *#connecttheleft* (a hashtag designed to connect the Democrats). Used together, these hashtags carry no information on the user's opinion. However, a human annotator would be able to identify the opinion by reading the entire text of the tweet. Hence, in order to augment the information obtained by using hashtags alone, we incorporated information from the tweet as well.

For this purpose, we used the n-gram model, which is considered a powerful tool for sentiment extraction [149]. n-grams are essentially contiguous sequences of n words extracted from text. The n-gram model was developed as a probabilistic language model which predicts the occurrence of the next word in the sequence of words by modeling it as an $(n-1)$-order Markov process. In the domain of sentiment analysis, n-grams have been widely used since

they help to capture phrases that carry sentiment expression [150, 67].

We begin by using hashtags separately as features in the logistic regression model (as described further), and then use them in conjunction with n-grams to achieve better results.

Popular hashtags: To eliminate the need for manual selection of hashtags, we extracted the most frequently used hashtags separately from each of the filtered datasets, by computing the total number of times each hashtag occurred in the respective dataset. For both the datasets, we used the 1000 most frequently used hashtags. We refer to these hashtags as ***popular hashtags***. Not surprisingly, a manual inspection revealed that all of the popular political tags were related to politics either by representing names of the parties, their representatives, or political issues that gained importance during that time period. A similar pattern was observed for the popular Obamacare hashtags.

We then used the frequency of use of the popular hashtags as features in our model. Thus, in equation (3),

$$X_j^i = \text{ number of times popular hashtag } j \text{ is used by user } i. \tag{3.5}$$

Popular n-grams in conjunction with hashtags: As discussed previously, we used n-grams to augment the hashtag information. We used values of $n = 1, 2$ to extract unigrams and bigrams from the tweets of each labeled user. Again, we picked the most popular n-grams from each dataset. For each dataset, we chose 2000 most popular unigrams and 2000 most popular bigrams. We combined the information we obtained from the hashtags with that obtained from the n-grams.

We tested each *type* of n-gram feature separately with the hashtags. Thus, when using

hashtags and unigrams as features, $X_{i:}$ is of size 1×3000 where

$$X_j^i = \text{number of times popular hashtag } j \text{ is used by user } i \text{ for } j < 1000$$

$$X_j^i = \text{number of times popular unigram } j \text{ is used by user } i \text{ for } j > 1000$$

3.5 Experimental Results

In this section we outline in detail our implementations of the proposed method with both l_1 and l_2 regularization, and the metrics we used to evaluate the results. Further, we describe the existing methods that we chose for comparison, and report the results obtained.

3.5.1 Experiments using Different Feature Sets

To evaluate the performance of the model, we conducted hold-out cross validation by randomly splitting the data into 30% test set and 70% training set. On each run of the cross-validation, the best λ was learned from the validation error on the training set. The cross-validation was done 10 times, with the data being randomly shuffled each time. Our experiments showed that the best λ value did not vary much across the validation sets of the respective dataset. For the U.S. Politics dataset, we set λ = 50.0 for l_2-regularization, and for l_1-regularization, it was 0.01. For the Obamacare dataset, we set λ = 25.0 for the l_2-regularized model, and λ = 0.0083 for the l_1-regularized model. The average classifier metrics [151] such as ROC curves, AUC, accuracy, precision, recall, F1-score and specificity across the 10 sets is reported in Section 5.4. For the U.S. politics data, we tested on 147 users, and on 60 users for the Obamacare dataset. For each user, the class with the higher probability is assigned as the corresponding opinion label, with ties broken arbitrarily. There were no cases in either of the datasets in which ties were encountered.

Table 3.2 presents the results obtained using logistic regression with l_2 and l_1 regularization

Table 3.2: Classifier metrics on U.S. Politics dataset, using l_2 and l_1 regularization. The features selected are described in Section 3.4.4. The 3rd column represents the number of features selected by the regularizer out of the total number of available features in parentheses. The best results are in bold.

Feature type	**Regularization**	**Number of Selected Features**	**Mean Accuracy**	**Mean AUC**	**Mean F1-score**	**Mean Specificity**
Hashtags	l_2	288 (1000)	$86.32(\pm 0.043)$	0.915	0.85	0.875
	l_1	22 (1000)	$84.70(\pm 0.048)$	0.896	0.823	0.82
Hashtags, unigrams	l_2	1488 (3000)	$86.12(\pm 0.031)$	0.896	0.843	0.885
	l_1	34 (3000)	$85.67(\pm 0.025)$	0.903	0.818	0.86
Hashtags, bigrams	l_2	1398 (3000)	$\mathbf{87.35(\pm 0.029)}$	**0.909**	**0.858**	**0.895**
	l_1	32 (3000)	$86.10(\pm 0.030)$	0.916	0.844	0.849
Hashtags, unigrams, bigrams	l_2	2430 (5000)	87.10 (± 0.027)	0.905	0.855	0.893
	l_1	70 (5000)	85.03 (± 0.033)	0.909	0.832	0.869

Table 3.3: Classifier metrics on Obamacare dataset, using l_2 and l_1 regularization. The features selected are described in Section 3.4.4. The 3rd column represents the number of features selected by the regularizer out of the total number of available features in parentheses. The best results are in bold.

Feature type	**Regularization**	**Number of Selected Features**	**Mean Accuracy**	**Mean AUC**	**Mean F1-score**	**Mean Specificity**
Hashtags	l_2	445 (1000)	$77.33(\pm 0.0466)$	0.912	0.804	0.799
	l_1	34 (1000)	$79.33(\pm 0.0466)$	0.90	0.824	0.83
Hashtags, unigrams	l_2	2295 (3000)	$87.30(\pm 0.022)$	0.942	0.906	0.943
	l_1	210	$86.50(\pm 0.022)$	0.920	0.891	0.91
Hashtags, bigrams	l_2	1506 (3000)	$87.54(\pm 0.025)$	0.956	0.907	0.927
	l_1	132 (3000)	$89.6(\pm 0.025)$	0.973	0.93	0.94
Hashtags, unigrams, bigrams	l_2	3448 (5000)	$\mathbf{90.8}$ $(\pm \mathbf{0.033})$	**0.958**	**0.919**	**0.850**
	l_1	372 (5000)	$89.1(\pm 0.030)$	0.945	0.88	0.825

on U.S. Politics, while the results for the Obamacare dataset are illustrated in Table 3.3. Four sets of features were used for both sets of experiments, and the best results are indicated in bold. As can be observed, the values of each of the classifier metrics are excellent. The high values of precision and specificity indicate that the method can predict both positive and negative opinions accurately. The highest accuracy achieved by our classifier was **87.35%** on U.S. Politics and **90.8%** on Obamacare.

It is to be noted that, using l_1 regularization, comparable accuracies were obtained with a much smaller number of features. For instance, for the U.S. Politics dataset (Table 3.2), using the combination of hashtags and bigrams, we were able to achieve a high accuracy of **86.10%** and an AUC of **0.916** from 32 features, as contrasted with using 1398 features and obtaining slightly higher accuracy of 87.35% and an AUC of 0.909 with l_2 regularization. A similar trend in results was observed for the Obamacare dataset as well.

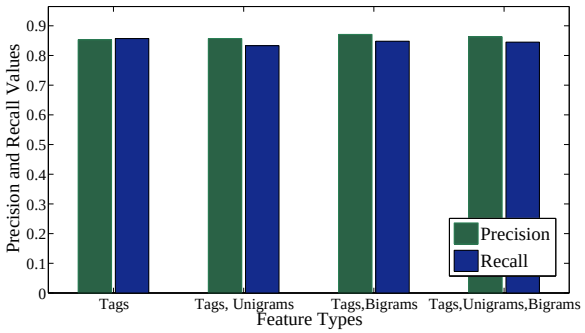
Figures 3.1(a) and (b) present the results obtained using logistic regression with l_2 regularization on U.S. Politics and Obamacare, respectively, and Figure 3.1(c) presents the ROC curves obtained using the l_2-regularized model on U.S. Politics and Obamacare.

Selection of Informative Features: From Tables 3.2 and 3.3, we found that the l_1 regularizer yields excellent results with a small number of selected features. Table 3.4 shows a few of the features that the regularizer picked from either dataset as the most informative features. Thus the method results in automatic selection of the most useful features for opinion detection.

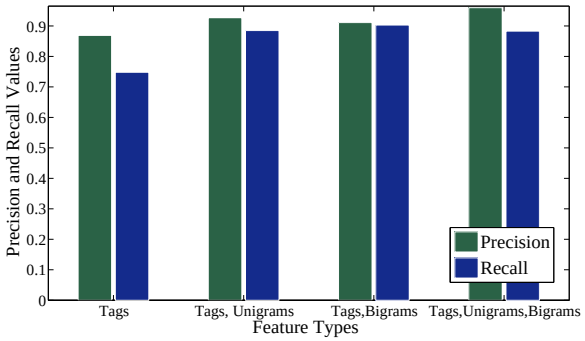
3.5.2 Comparison with existing methods

We compare our methods with three popularly used state-of-the-art methods that perform tweet-level sentiment analysis, and use their results to obtain opinions on a user level as described below. The following methods were tested on the U.S. Politics dataset.

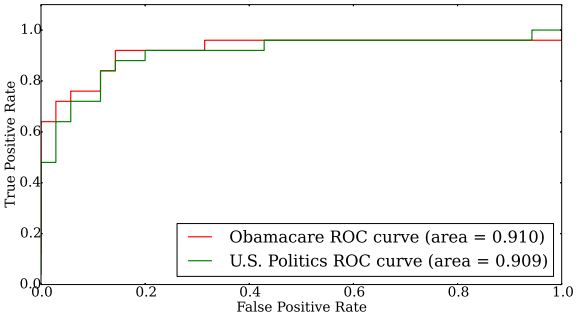
SentiStrength: SentiStrength [59] is a lexicon-based method that was designed for use with



(a) Precision Recall values for U.S. Politics



(b) Precision Recall values for Obamacare



(c) ROC curves for U.S. Politics and Obamacare

Figure 3.1: Classifier metrics with l_2 regularization

Table 3.4: Examples of features selected by l_1-regularization

Feature Type	**Dataset**	**Sparse features**
Hashtags	U.S. Politics	*tcot*, *p2*, *gop*, *obama2012*
Bigrams	U.S. Politics	*tcot gop*, *obama didnt*, *mitt romney*
Hashtags	Obamacare	*obamacare*, *tcot*, *defundobamacare*, *defund*
Bigrams	Obamacare	*defund obamacare*, *shut down*, *government over*

short informal text including abbreviations and slang. It has been widely used by researchers for sentiment analysis of tweets, online posts, etc. (Section 3.2). It uses a lexicon of positive and negative words which were initially annotated by hand, and later improved during a training phase. Given a sentence, the method assigns a sentiment score to every word in the sentence, and thereafter, the sentence is assigned the most positive score and the most negative score from among its words. According to [59], the algorithm was tested extensively for accuracy, and was found to outperform standard machine learning approaches. Hence we chose this as a baseline method to compare against.

Tweet-level Maximum Entropy Classifier: The second method for comparison is a machine-learning method proposed in Section 3.3 of [51] which uses a Maximum Entropy based classifier trained on 1,600,000 tweets using emoticons as noisy labels. It uses the presence or absence of unigrams, bigrams and parts-of-speech tags as features for classification, and classifies a given tweet as positive or negative. The authors provide an online tool for this purpose [152], which we use for conducting our experiments. This method has also been widely used for sentiment analysis. It is to be noted that we used their pre-trained model that was trained on their annotated tweet set. We could not train the method on our labeled datasets because our datasets have labels on the user and not on the individual tweets, and it is non-trivial to transfer the user opinion to their tweets owing to the amount of noise per tweet. Moreover, we could

Table 3.5: Comparison of the proposed method with three state-of-the-art methods

Method	Accuracy(%)	Precision	Recall	Specificity
l_2 - regularized Logistic regression	**87.35**	**0.871**	**0.848**	**0.895**
SentiStrength	53.06	0.485	0.586	0.485
Maximum Entropy method	44.29	0.525	0.419	0.463
Combined method (SentiStrength and MaxEnt)	59.59	0.542	0.694	0.515

not annotate *our* datasets using emoticons because they are rarely used in our datasets (only 0.13% of the tweets used emoticons in the U.S. Politics dataset). Since the authors [51] used emoticons to label the sentiment of a tweet and did not manually annotate them, theirs may be considered as a partially supervised method, as opposed to our fully supervised method.

Combined Method: The third method for comparison is a method described in Section 3.2 of [143] that combines the output of the lexicon-based method [59] and the tweet-level machine learning method [51]. The authors propose a way to combine the results of SentiStrength and the MaxEnt based method of [51] to perform a binary tweet-level sentiment classification with better accuracy than either of the individual methods.

Obtaining targeted user-level sentiment from tweet-level sentiment: We adopt the following strategies when comparing our method with the other three methods. First, to obtain a sentiment label for every tweet using *SentiStrength*, the most positive and most negative scores for every tweet were added up. If this sum was positive the tweet was labeled positive; if the sum was negative then it was labeled negative, and if the sum was zero the tweet was labeled neutral. This approach was proposed in Section 3.2 of [143].

Second, all of the methods described above determine whether a given tweet has an *overall positive or negative sentiment, irrespective of the target of the sentiment*. This varies from our definition of *positive* and *negative* as described in Section 3.4.1. Hence, to determine the sentiment of a tweet towards a target (Democrat or Republican), we selected a set of keywords that were associated with Democrats and another set for Republicans, with the objective of

identifying targets for as many tweets as possible, and defined them as *positive targets* and *negative targets*, respectively. (The keywords used are given in Table 3.1). For **any** method that we compared with, given a tweet sentiment, we first computed a sum of the target words that the tweet contained, assigning +1 for a *positive target* and -1 for a *negative target*. If the sum was greater than 0 we assumed that the subject of the tweet was Democrats, in which case the sentiment remained unaltered. If the sum was less than 0 we assumed that the subject was Republicans. In this case, a positive sentiment towards Republicans would mean a *negative* sentiment according to our definition, and vice versa.

Third, to obtain *user-level* sentiment from the tweet-level sentiment output from **any** of the methods, we adopted the following strategy. For every user, we summed the (targeted) sentiments of all her tweets using +1 for *positive*, -1 for *negative* and 0 for neutral. The user output was considered positive if the sum was positive, negative if the sum was negative and was assigned randomly if the sum was zero. Table 3.5 represents the comparison of our method with the existing methods. All of the classifier metrics clearly display that our method outperforms the other methods.

3.6 Conclusion

We proposed a method for detecting user-level opinion on a given topic from Twitter data. Our approach of performing user-level (as opposed to tweet-level) opinion detection using regularized logistic regression with hashtags and n-grams as features was found to produce excellent results. The l_2 and l_1 regularizations yielded comparable accuracy, however the l_1 regularization required far fewer features. Moreover, our method required no manual labeling of features. The method was applied to Twitter datasets on two different topics and yielded excellent results on both, which highlights its generalizability. The importance of informative features is evident in the results obtained; only a small percentage of the most informative

features were required for accurate user opinion detection.

Chapter 4

Detecting Opinions in a Temporally Evolving Conversation on Twitter

4.1 Introduction

As has been previously discussed in Chapter 1, there is an inherent temporal aspect to user-generated online textual content, since, on most topics, users continue to post their views over time. In such a scenario, opinion detection over time becomes a necessity. In this Chapter, we present a method for detecting the opinions of Twitter users on a given topic over time, using data mining and machine learning techniques. Methods in the field of opinion detection in general are based either on machine learning, or lexicons of words (discussed in detail in Chapter sec: opinion mining overview). There is no temporal aspect to these approaches. They are trained on labeled data and/or use a pre-determined lexicon of words. However, in the case of temporal opinion detection, which is the problem we address here, the focus of the conversation shifts from one sub-topic to another, thus new textual features emerge at every time point. The lack of training data at every timestep renders general supervised approaches infeasible.

The method we propose in this work for temporal opinion detection borrows from social network research conducted by sociologists over the years [90, 91]. A key observation from social network research is that temporal evolution of user opinions is a slow process. People are inherently resistant to changing their opinions. We propose a regularized supervised approach that requires training only at the initial time, and enables us to use opinions detected in a previous timestep when performing predictions for the future. Additionally, the method can capture relevant textual features over time, thus highlighting the conversational sub-topics that emerge at every timestep.

We selected Twitter as the source of data for our experiments, and Obamacare as the primary topic of interest. Obamacare is a popular term coined to represent the Affordable Care Act (ACA) which was signed into law by President Barack Obama on March 23, 2010 [153]. Since its inception, it has garnered much political and social attention in the US, and has emerged as one of the most popular topics of discussion in social media platforms [154]. The Act also underwent several reforms over time, each addressing a different issue. This led to an evolving online conversation on the topic, since the focus of the discussions would shift from one sub-topic to another over time. The above characteristic made this topic interesting and challenging for opinion detection, as we shall illustrate in the later sections. In order to demonstrate the generality of our method, we selected another topic for our experiments, namely, the U.S. Immigration Reform bill (the Border Security, Economic Opportunity, and Immigration Modernization Act of 2013) that was introduced in the US Senate in April, 2013. The bill would allow for many undocumented immigrants to gain legal status and become U.S. citizens. Additionally, it would make the border more secure by adding up to 40,000 border patrol agents [155]. This topic was also extensively discussed on Twitter. Details of the data collection process for both topics are elaborated in Section 4.4.1.

Contributions of this work: The contributions of this work are as follows:

1. This work proposes a machine-learning model to accurately detect opinions of Twitter users over time using their tweets, even when the topic of conversation is evolving in nature. Training is required only at the initial time.

2. The proposed method also showcases the textual features that are most effective at identifying the opinions at different time points. These features aid in identifying the most popular sub-topics that emerge at every time point.

4.2 Related Work

Publicly available sentiment lexicons such as SentiStrength [59] and SentiWordNet [60, 61] have been extensively used for sentiment analysis of social network data, online posts, movie reviews, etc. [64, 63, 59, 144]. However, as seen in our previous work (Chapter 3) [43], these do not perform well for opinion detection on Twitter users. Further, prior sentiment analysis methods using Maximum Entropy, Naive Bayes, SVM [51], k-NN based strategies [141], label propagation [142], etc. address the problem of temporal opinion detection that is the topic of this paper.

In prior work (Chapter 3) [43], we addressed the problem of opinion detection of Twitter users over a fixed period of time. There was no temporal aspect to the problem. We developed a supervised learning approach using a regularized logistic regression model. We used textual features, namely hashtags and n-grams, to detect user opinions on two topics: U.S. Politics and Obamacare, with a high accuracy. The Obamacare dataset used in that work contained tweets over a short time period and hence did not capture the evolving nature of the conversation. However, when we applied the same method to the current dataset that spans a larger timeline, it failed to detect user opinions accurately (details in Section), thus leading us to the development of the proposed model for temporal opinion detection.

4.3 Temporal Opinion Detection over an Evolving Conversation

In this section we describe the problem at hand and discuss the social network research that our model is based on. Thereafter, we delve into the details of the model.

4.3.1 Opinion Change Processes Over Time - *The Basis of our Model*

The key point of our opinion detection model is that users tend to change their opinions very slowly. This forms a basis of the seminal opinion change models from sociology [91, 90]. We present three factors owing to which transition to a different opinion takes place gradually. First, people vary in their readiness to be influenced by their neighbors. Every person has some amount of stubbornness and attachment to their own opinions and beliefs. This is a factor that most models of opinion change consider. For example, a widely-used opinion change model arises from the Social Influence Network Theory of Friedkin and Johnson [91], and is given by

$$\mathbf{y}^{(t)} = \mathbf{AW}\mathbf{y}^{(t-1)} + (\mathbf{I} - \mathbf{A})\mathbf{y}^{(1)}, \tag{4.1}$$

where $\mathbf{y}^{(t)}$ is a vector of the users' opinions at time t, and $\mathbf{W} = [w_{ij}]$ is the matrix of interpersonal influences, which stores the amount of influence user j has on user i. $\mathbf{A}$ is a diagonal matrix of the users' susceptibilities to interpersonal influence. As is evident from (4.1), $\mathbf{A}$ determines how anchored the users remain to their initial opinions $\mathbf{y}^{(1)}$, which regulates how much they are influenced by their network neighbors to change their opinions.

Second, we treated the responses of all users as homogeneous from the point of view of opinion change. Thus the opinion of any user, as well as the opinions of all the users she is influenced by, evolve over time. The influenced user slowly changes her opinion in response to the changing opinions of her influencers.

Third, multiple neighbors influence each user. Most opinion models, including Social Influence Network Theory (4.1) and the DeGroot model [90], assume that a user's opinion is the average of the opinions of her neighbors and her own opinions. This averaging effect tends to dampen dramatic changes [91], making opinion change a slow process. This key observation leads to the main assumption in our new model. *For a sufficiently large set of users, most users are not likely to change their opinions drastically over a short period of time.*

4.3.2 Temporal Opinion Detection Model

In previous work [43], we assumed user opinion to be a distribution over positive and negative types, and used textual features derived from tweets to learn a weighted combination of the features that would best classify the opinions (3.3). In this work, we extend the previous regularized logistic regression model, with an added element of time. As in the previous work, user opinions are classified as positive and negative types. Here, we have data samples $\mathbf{x}_i^{(t)}$, $i = 1, ...n$ and $t = 1, 2, ..., T$. Further, we have labels only for the first timestep, i.e., $y_i^{(1)}$, $i = 1, ...n$. Labeled samples are required for the first timestep, but not for the subsequent timesteps. Now, extending (3.1) for any t^{th} timestep for user i, we obtain

$$P(y_i^{(t)} = 1|\mathbf{x}_i^{(t)}, \beta^{(t)}) = \frac{1}{1 + \exp(-\beta^{(t)^T}\mathbf{x}_i^{(t)})}, \tag{4.2}$$

where $y_i^{(t)}$ is the discrete opinion value in $\{-1, 1\}$ in timestep t, $\mathbf{x}_i^{(t)}$ is a $k \times 1$ data vector and $\beta^{(t)}$ is a $k \times 1$ feature weight vector for timestep t.

We do not have labels on the samples for timestep $t + 1$, as previously stated. Hence, to predict the opinions for timestep $t + 1$, we apply the key observation from Section 4.3.1 that *most* users do not change their opinions drastically in a single timestep. Thus, we assume that *most* users hold the same opinion as in the previous timestep. Most of the opinions in the previous timestep will therefore be the same as those in the next timestep, i.e. $y_i^{(t)}$ is the same

as $y_i^{(t+1)}$ for most users. Following this assumption, we use $y_i^{(t)}$ from the previous timestep, and new textual features $\mathbf{x}_i^{(t+1)}$ from the current timestep to learn $\beta^{(t+1)}$.

Thus, we minimize the following l_2-regularized logistic loss function over consecutive timesteps t and $t+1$:

$$L(\beta^{(t+1)}) = -\log\left(\prod_{i=1}^{n} P\left(y_i^{(t)}|\mathbf{x}_i^{(t+1)}, \beta^{(t+1)}\right)\right) + \lambda\|\beta^{(t+1)}\|_2^2 \tag{4.3}$$

$$= \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i^{(t)}({\beta^{(t+1)}}^T \mathbf{x}_i^{(t+1)})\right)\right) + \lambda\|\beta^{(t+1)}\|_2^2 \tag{4.4}$$

The regularization helps to avoid overfitting [136] and to take care of the fact that this is an underdetermined system since $n << k$. Thus, by minimizing (4.3), we learn $\beta^{(t+1)}$ even in the absence of labeled samples at time $t+1$. We use the open-source machine learning tool scikit-learn [148] to implement logistic regression with l_2 regularization.

4.4 Data Collection and Preprocessing

In this section we describe the method used to collect the dataset for this work, and the data pre-processing steps involved.

4.4.1 Data Collection

We selected Twitter as the source of data for our experiments, and Obamacare as the primary topic of interest. *Obamacare* is a popular term coined to represent the Affordable Care Act (ACA) which was signed into law by President Barack Obama on March 23, 2010 [153]. Since its inception, it has garnered much political and social attention in the US, and has emerged as one of the most popular topics of discussion in social media platforms [154]. The Act also underwent several reforms over time, each addressing a different issue. This led to an

evolving online conversation on the topic, since the focus of the discussions would *shift* from one sub-topic to another over time. The above characteristic makes this topic interesting and challenging for opinion detection, as we shall illustrate in the later sections.

In order to demonstrate the generality of our method, we selected another topic for our experiments, namely the U.S. Immigration Reform bill (the Border Security, Economic Opportunity, and Immigration Modernization Act of 2013) that was introduced in the US Senate in April, 2013. The bill would allow for many undocumented immigrants to gain legal status and become U.S. citizens. Additionally, it would make the border more secure by adding up to 40,000 border patrol agents [155]. This topic was also extensively discussed on Twitter.

To crawl tweets on a topic of interest, we randomly selected users and collected their tweets over a period of time using the Twitter Streaming API [156]. For Obamacare, tweets were crawled over a period of 8 months from July 2013 to February 2014. We have 757,960 users and 4,203,900 tweets in our dataset. For the topic of Immigration, tweets were crawled over the months of July, August and September, 2013, yielding a total of 15,001 users and 44,626 tweets. We consider each month to be 1 timestep for the sake of our experiments. On the topic of Obamacare, we selected 936 users that have tweets every month on which to test our model, and for the topic of Immigration, we picked 111 users.

For data pre-processing, we used the steps described in Chapter 3.3.2.

4.5 Implementation Details

In this section we describe the features we chose to use in the model, and also explain the steps taken to implement the model.

4.5.1 Feature Engineering

As in our previous work (Chapter 3) [43], we used hashtags and n-grams as features for our model. At every timestep, we ordered the features according to the number of users that use them. We use the 1000 most popularly used hashtags, 2000 most popularly used unigrams and 2000 most popularly used bigrams from each timestep for our experiments. The choice of the number of features was governed by the usage of the features. For instance, after the first 1000 hashtags, the usage of the hashtags drops significantly, thus motivating us to use the most popular 1000 tags as our features. Similar reasons led to the use of the top 2000 unigrams and bigrams. Thus we had 5000 features at every timestep.

For every user i at time t in (4.3), $\mathbf{x}_i$ contains the number of times user i uses each of the 5000 features at that timestep. Owing to the evolving nature of the conversation, this set of features changes over time. However, using our model described in Section 4.3.2, we can *automatically* learn a new β at every timestep for a new set of features by minimizing (4.3). Tables 4.1 and 4.2 show a few examples of features found on several timesteps.

4.5.2 Implementation

In our experiments, we considered each month to be a timestep, and studied the same set of n users across all timesteps. The following provides a detailed description of the steps taken at every timestep.

- At timestep 1:
  - We begin by labeling a subset of the users such that those with a positive opinion on the given topic are assigned a label +1 and those with a negative opinion are assigned a label -1. Let d be the number of users that are labeled at timestep 1. The

Table 4.1: Examples of hashtags and n-grams over time on Obamacare

Feature type	**Timestep 1**	**Timestep 5**	**Timestep 8**
Hashtags	#obamacare, #koch, #getcovered, #cvs, #gop	#obamacare, #fullrepeal, #dontfundit, #aca, #trainwreck	#obamacare, #irs, #koch, #debtceiling, #gop
Unigrams	obamacare, gop, health, republicans, healthcare	obamacare, website, insurance, fix, coverage	obamacare, enrollment, work, hhs, job
Bigrams	obamacare will, the gop, benefits to, howard dean, fund obamacare	obamacare enrollment, signed up, fix obamacare, website failed, obamacare promises	3.3 million, signed up, million jobs, the koch, the irs

Table 4.2: Examples of hashtags and n-grams over time on Immigration

Feature type	**Timestep 1**	**Timestep 3**
Hashtags	#immigration, #takeittothehouse, #weallshallovercome, #moveforward, #immigrationenforcement	#immigration, #immigrationnews, #protests, #deport
Unigrams	immigrants, taxes, system, reform, drafted	gop, population, reforms, senator
Bigrams	million people, to diversity, immigration reform, require immigration	gop is, for immigration, need jobs, domestic issue, immigration reform

data matrix is built using the 5000 textual features (as described in Section 4.5.1), thereby leading to a $d \times 5000$ matrix, $X^{(1)}$. We use this data to train the model (4.3) to learn $\beta^{(1)}$. For Obamacare, $d = 201$ (89 positive, 112 negative), and for the Immigration dataset, $d = 30$(24 positive, 6 negative).

- We then assign opinion labels to the larger unlabeled set of $n - d$ users using the learned $\beta^{(1)}$. This step is performed to get the opinion labels for all n users at this timestep. We now proceed with the entire set of n users for the subsequent steps.

- For each subsequent timestep, $t + 1$:

  - We minimize the regularized logistic loss function (4.3) between the opinions of users at t and $t + 1$ to learn $\beta^{(t+1)}$.

  - We then use the learned $\beta^{(t+1)}$ to predict opinions at time $t + 1$. This forms $y_i^{(t+1)}$.

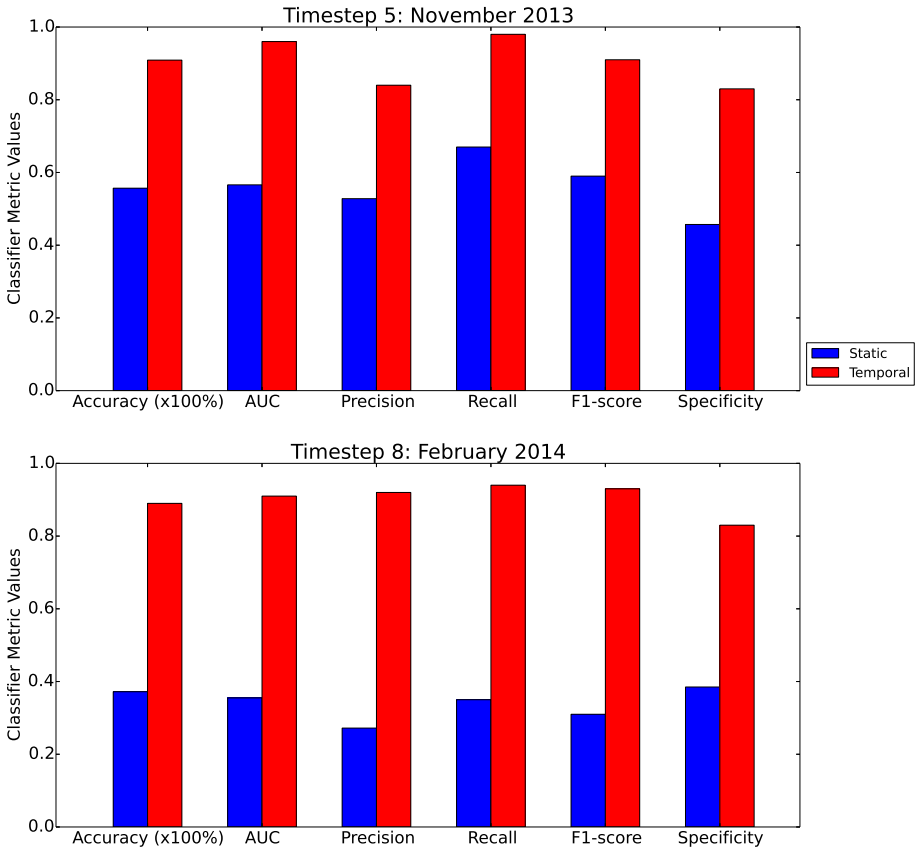


Figure 4.1: Comparison of the static and temporal opinion detection methods on Obamacare. The methods are compared on two timesteps of interest across all classifier metrics.

4.6 Experimental Results

In this section, we outline in detail the experiments we conducted on the dataset, and the metrics we used to evaluate it. Further, we report the insights that the method provided with respect to the sub-topics that were being discussed at every timestep.

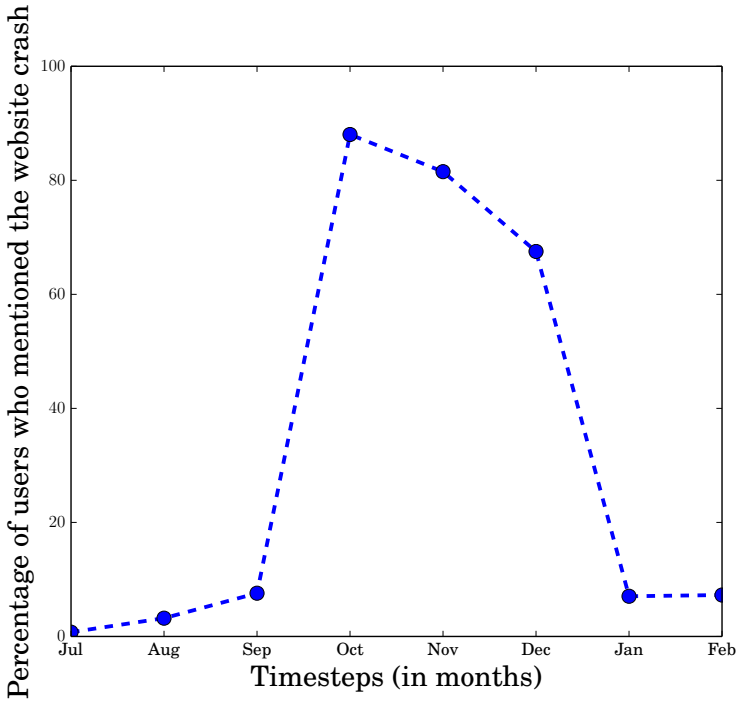


Figure 4.2: Mentions of Obamacare website crash over time

4.6.1 Temporal opinion detection results

To evaluate the model on our primary topic of interest, *Obamacare*, we labeled the opinions of a random group of users on some of the key timesteps to test whether our model captures their opinions correctly. We were particularly interested in determining whether the model detects the opinions correctly after the occurrence of a significant event with respect to Obamacare. One such event occurred on October 27, 2013, when the main website for the Affordable Care Act, *Healthcare.gov* crashed. This created a great deal of chatter on Twitter (see Figure 4.2 for a plot of the number of users that mentioned the website crash over time. As is evident, the number of users goes up significantly towards the end of October which was when the website crash occurred, and continues to be a focus of conversation during November as well.) To determine whether our model captures the opinions being echoed right after this occurrence, we focus on Timestep 5 which contains tweets from the beginning of November

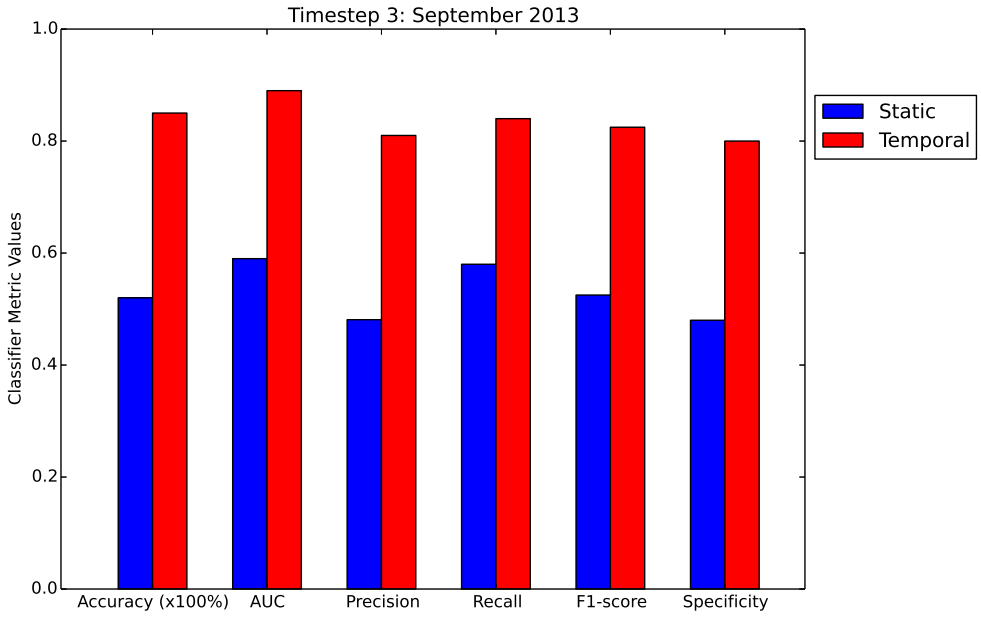


Figure 4.3: Comparison of the static and temporal opinion detection methods on Immigration. The static method is no better than random guessing after 2 timesteps, but the temporal method shows high predictive power.

2013, and throughout the rest of the month. We select 88 users at random from that timestep, for testing our model.

The other timestep that we picked for these tests was Timestep 8, which was the month of February 2014. In that month, the Department of Health and Human Services (HHS) announced the signing up of 3.3 million people for Obamacare, which was a significant event in the Obamacare timeline. Another event that generated a large volume of tweets at that time was that some firms were firing employees to avoid Obamacare costs, but were certifying to the IRS that the firings were not on the grounds of Obamacare, to avoid penalty of perjury. We labeled 43 randomly selected users from this timestep.

To validate the usefulness and the need for our method, we first present the results obtained by simply using the *Static Opinion Detection Model* described in Chapter 3.4.3, for temporal

opinion detection. Thus, in this case, we would not learn a new $\beta^{(t)}$ at every timestep t, but would use the β learned from the training samples at timestep 1 to predict opinions for later timesteps. We report the results obtained by using both the static and the proposed temporal models in Figure 4.1. As can be observed, the accuracies achieved using the static method on timesteps 5 and 8 are 55.68% and 37.2% respectively, while our new temporal method yields accuracies of 90.9% and 89.0% respectively for the two timesteps. Moreover, the temporal method outperforms the static method across all popularly-used classifier metrics [151] such as AUC, F1-score, etc.

To demonstrate the generality of our method, we also conducted experiments on the topic of *U.S. Immigration Reform bill*. Since we only have 3 months' data on the topic, we evaluated the classifier metrics on the last month. The results are reported in Figure 4.3. The temporal method yields better performance than the static method in this case as well. As is evident, the static method yields about 50% accuracy, which can simply be obtained by random guessing. However, using the temporal method yields a significantly higher accuracy of 85%. The temporal method also performs much better in comparison to the static method across all classifier metrics as well.

4.6.2 Significant feature detection and emergence of temporal sub-topics

Out of the 5000 features used at every timestep, some of the textual features are more informative in detecting opinions than others. To determine this set of informative features over time, we evaluated the statistical significance of each feature of the Obamacare dataset for predicting user opinion. We followed the technique described in Section 5, Algorithm 3 of [157] for significance testing, which we describe here for the sake of completeness.

For the timestep of interest, we ran our l_2 regularized temporal model on the data, and stored the weights that each of the features are assigned by the model. Then we randomized

the labels on the samples and ran our model on the randomized data. Let $\hat{\theta}$ be the coefficient obtained from this set. For each randomized run m, let $\tilde{\theta}$ be the random coefficient vector obtained from the fixed feature vector $\mathbf{x}$ and the randomized response $\tilde{y}$. For ν randomized runs, we obtained ν coefficient vectors $\tilde{\theta}$. For each dimension, the coefficient value in each $\tilde{\theta}$ represents a random statistical relationship between the feature and the response. Then the p-value of the l^{th} dimension is computed as

$$\frac{\text{Count}(|\tilde{\theta}_l| > |\hat{\theta}_l|)}{\nu + 1} \tag{4.5}$$

where "Count" represents the number of times the absolute value of the random coefficient for the l^{th} dimension exceeded the absolute value of the same coefficient obtained from the training set. This is a commonly used permutation test for statistical hypothesis testing [158]. Features that had a p-value less than 0.05 were selected as the most significant features with a confidence of at least 95%.

Using the significant features obtained at each timestep, we examined the dataset for tweets carrying these features. This led to the discovery of the various sub-topics of conversation (related to the main topic of Obamacare), that users participated in over time. Most of the sub-topics can be tied to real-world events that aligned with the timestep under consideration. This further reflects the evolving nature of the topics of conversation.

Tables 4.3, 4.4, 4.5 illustrate the sub-topics of interest that were detected over the various timesteps. For example, in July 2013, the IRS emerged as an important sub-topic of discussion. Similarly, Obama's apology and a count of how many people were enrolling in Obamacare were popular sub-topics in November 2013. In February 2014, the 3.3 million enrolment mark and Megyn Kelly (a Fox News anchor who covered a great deal of negative news related to Obamacare) were sub-topics that emerged as being popular. Thus our method is successfully able to detect evolving sub-topics of conversation among users over time.

Table 4.3: Significant features (95% statistical significance) on Obamacare at in July 2013. Significant features capture the temporally evolving sub-topics.

Time step	**Significant Features**	**Temporal sub-topics inferred from tweets**
Jul 2013	*braveheart, gifs*	The Washington Examiner publishes funny series of gifs from movie Braveheart depicting Republicans' failed attempts at defunding Obamacare.
	employees	News sources report that Obamacare call center employees were not being offered healthcare benefits.
	kyle	News report by reporter Kyle Cheney on Politico.com stating that CVS was going to publicize Obamacare.
	irs	IRS employees unwilling to sign up for Obamacare, although IRS was heavily involved in enforcing Obamacare.
	howard	Howard Dean, former Democratic National Committee Chairman, comments that Independent Payment Advisory Board will be unable to keep costs down.
	premiums	Obamacare premiums are lowered even further in eleven states.
	empire	Cited article discussing civil lawsuits, environmental damage caused by the output from industries, etc. of the Koch brothers' empire and related controversies.

Table 4.4: Significant features (95% statistical significance) on Obamacare in November 2013.

Time step	**Significant Features**	**Temporal sub-topics inferred from tweets**
Nov 2013	*warning*	Republicans "warning" people of Obamacare, and that the website crash is a "warning" in itself.
	case	Blog by Peter Suderman ("Time To Start Considering Obamacares Worst-Case Scenarios") discussing failure of online enrollment system negatively affecting Obamacare.
	apology	• Obama apologizing to people whose insurance plans were being canceled, even though he said that people could keep their existing coverage if they liked. • Ed Schultz demands that Republicans, rather than the President, should apologize "for not having any plan".
	scorecard	Obamacare scorecard: how many actually enrolled, and how a larger number of people lost their insurance.

Table 4.5: Significant features (95% statistical significance) on Obamacare in February 2014.

Time step	**Significant Features**	**Temporal sub-topics inferred from tweets**
Feb 2014	*@megynkelly*	Megyn Kelly, a Fox news anchor who covered (negative) news related to Obamacare.
	wednesday	Dept. of Health and Human Services announces on a Wednesday (Feb 12, 2014) that 3.3 million people signed up for Obamacare, but it includes hundreds of thousands of individuals defaulting their first premium payment.
	firings	Firms required to certify to the IRS that Obamacare was not a factor in their firing their employees (although it was).
	tgdn	New hashtag (Twitter Gulag Defense Network) started in January 2013 to counter Twitter Gulag, a way to trick Twitter systems into thinking that live profiles are actually spambot profiles. Apparently, many conservative profiles were being shut down by leftists employing this policy.

4.7 Conclusion

In this work, we proposed a novel temporal opinion detection method that could successfully detect the opinions of Twitter users engaging in an evolving conversation. Our primary topic of interest was Obamacare, for which the focus of conversation shifted from one sub-topic to another due to the various events associated with the event that occurred over time. We also selected the topic of U.S. Immigration Reform to demonstrate the generality of our method. Our proposed temporal machine-learning method performs well across all classifier metrics of importance. Additionally, it leads to automatic detection of informative features that point to important, and changing sub-topics.

Chapter 5

Mining Aspects and Opinions on Large Scale Review Data using Distributed Representation of Words

5.1 Introduction

With the accessibility and widespread use of the Internet in general, and the rise of social media in particular, user-generated textual content has become pervasive and user opinions are now available freely in the form of reviews on various websites, blogs and comments on social media. Online marketplaces such as Amazon, BestBuy etc. act as a rich source of consumer reviews on the products they sell. Similarly, Yelp and TripAdvisor host millions of reviews on restaurants, businesses, sights, hotels, etc. Consumer feedback is crucial for companies to understand how their products and services are perceived, how they fare in comparison with their competition and to help them improve their products and services when the next version is rolled out. Moreover, from the point of view of the consumer, comments and reviews are highly important since learning the opinions of others helps them in their purchase decisions.

Although we have access to large scale user-generated data today, much of the user-generated feedback is in the form of very noisy text from which it is difficult to extract information. Some of the key challenges of working with online text data are: the ambiguity inherent in natural language [133], extreme sparsity [134, 135] and the abundance of noise [42, 41]. Noise may include grammatical errors, misuse of punctuations, spelling errors etc. Individual NLP tasks such as spelling correction, stemming, lemmatization, POS tagging, etc. are often needed to capture signals from such noisy data, which unfortunately do not scale very well. In addition, specific domain understanding is often required to improve the performance of specific NLP algorithms [159, 160, 161].

To address these challenges, in this work we present a machine learning approach that utilizes Word2Vec [162, 163], a scalable neural network model that produces a vector space representation of words in order to provide accurate sentiment prediction and a human interpretable summary of user-generated online product reviews. The use of this method enables us to extract meaning out of noisy data without having to employ many of the NLP tasks mentioned above.

The motivation of our methodology comes from the following observation about user reviews. When users review a service or a product, not only do they express their overall opinion on the subject, but they also demonstrate their likes and dislikes over various attributes and functionalities of the service or product in question. For example, when assessing a restaurant, one might like or dislike the food quality, the ambience, the portion sizes and so on. The National Restaurant Association lists various factors that users consider when choosing a place to eat [164]. Thus, in order to effectively understand why a restaurant is worth eating at or not, it is important to understand these key drivers of sentiment. This leads us into the task of aspect-based sentiment analysis, one of the key frameworks of sentiment analysis today [46, 71, 47, 165]. In the present work, we aim at uncovering the key drivers of sentiment from reviews in an automated fashion, using distributed representations of words, i.e. Word2Vec

[162, 163]. We used a publicly available Yelp review dataset [30] for conducting our experiments. We specifically focused on restaurant reviews extracted from this dataset, although the method we propose could easily be extended to reviews on any topic, such as products, vacations, destinations, etc.

Reviews usually contain a numeric rating assigned by the consumer. This rating can be thought of as a mix of positive and negative sentiments that the user feels towards various aspects, details of which may occur in the review text. We used the ratings as labels to train a classifier in order to determine the sentiments associated with the key drivers. Our method performed well across all classifier metrics. Further, using the learned classifier coefficients, we were able to analyze reviews and understand aspects of the topic, i.e. restaurants, that contribute to user satisfaction or dissatisfaction.

Contributions of our work: The main contributions of our work are:

- We developed a method to identify the key aspects of restaurants that are reviewed online and capture the sentiment associated with them. Our method helps in obtaining structure and information from user-generated review data which is mostly comprised of noisy, unstructured text.

- Our method provides excellent coverage of the dataset by aggregating contextually similar words, thereby reducing feature space and data sparsity.

- Further, we present in-depth aspect-level analysis of the reviews along with comparative analyses on different kinds of restaurants.

- Although our experiments are conducted on restaurant reviews, the method is generalisable and can be applied to reviews on any service or product, as we exhibit in Appendix A.

5.2 Related Work

Research in the area of aspect-based sentiment analysis can be broken down into topic modeling based approaches and machine learning based approaches. The topic modeling based methods can be further categorized into two subsets - those that separate the task of discovering aspect and sentiment words [46, 47] and those that do not [71, 72]. [71] proposes a flat topic model based on LDA [79], in which a flat mixture of topics is associated with each polarity and all the words with this polarity are generated from this mixture. [47] uses a hybrid model based on Maximum-Entropy and LDA to separately uncover aspect and sentiment words. However, as stated in [166], fully unsupervised models often result in topics that are not always comprehensible by humans, owing to the fact that the objective function used in these topic models does not often correlate well with human judgement.

Outside of the topic modeling framework, Parts-of-Speech (POS) tagging is a widely used method for this problem. The methods proposed in [48], [49] and [73] apply POS tagging to identify nouns and noun phrases, based on the observation that aspects or features are generally nouns [167]. In particular, [48] uses association rules to identify frequent noun phrases, each of which is a possible aspect. In [49], aspects are extracted by computing pair-wise mutual information between noun phrases and a set of meronymy discriminators associated with the product category. Similarly, [73] uses POS tagging along with a language model approach which assumes that product features are mentioned more often in a product review than in generic English.

The above methods are different from ours since none of them use distributed representation of words, and hence do not capture the contextual similarity between words. However, since POS tagging is a popular method, we used it as a baseline to compare with. We elaborate on the baseline later.

Table 5.1: Examples of Reviews from the dataset. The words in bold indicate noise in the text. Noise includes mis-spellings, case insensitivity, misplaced punctuation marks etc.

Review	Rating
My favorite breakfast place. Have good sandwiches also. Stopped again for **Bfast** and had the mixed grill–get the small portion unless you are a real **MAN**! Mixed grill has sausage, (could it be Ricci's?), eggs, onions, and home fries, **soooo gooooooooood**! Use Mancini's bread for toast, got the raisin toast - Yum.	5.0
I was first introduced to this place by a friend which ended up being a location we'd frequent when we couldn't decide on where to go, or what to eat. This would be the place we'd hit up for breakfast and on Sundays they have a special brunch menu which offers different items and a buffet style course.	3.0

5.3 Dataset and Challenges

The dataset we used is a subset of the dataset provided by the Yelp Dataset Challenge [30]. The dataset contains reviews and ratings of businesses as provided by Yelp users, along with meta data consisting of the name and location of the business, the type of the business, etc. To obtain a dataset on a single topic, we extracted reviews pertaining to restaurants and thereafter, take a subset of that data. Each review consisted of the text of the review, along with the rating that the user provided for that restaurant, which ranges from 1.0 to 5.0. Table 5.1 shows some review examples from our dataset.

For the task of sentiment analysis, we used the numeric ratings as a way to label reviews as *positive* or *negative*. On exploring the data, we find that the reviews with ratings 1.0 and 2.0 are mostly negative towards the restaurant under review and those with ratings 4.0 and 5.0 carry positive sentiment. We labeled reviews with ratings 1.0 and 2.0 as negative, and those

with ratings 4.0 and 5.0 as positive. We find that reviews with ratings 3.0 are often ambiguous and hence we omit them as samples. Our dataset consisted of 611,696 reviews in all.

Further, we divided the entire dataset using stratified sampling into training (75%) and test (25%) data. This ensures that the rating distribution is retained in both sets. We used the training data for model training purposes, as will be subsequently discussed. The test data is used to evaluate our methodology.

The challenges we encountered for this problem have been discussed earlier in Section 1.4.

5.4 Outline of Methodology

We developed a methodology for automated extraction of key drivers of sentiment from review text, and leveraged these drivers in constructing features. These features were subsequently used in a machine learning model for identifying sentiment. In this section, we define and discuss a few key concepts, and present an outline of our proposed methodology.

5.4.1 Key Drivers of Sentiment

We aim to identify the aspects that users base their reviews on, as well as the sentiment associated with the aspects. Thus, we proposed the identification of the following two groups of words from the reviews:

- **Aspects:** Aspects are the features or attributes of the restaurant under review, such as *food, service, ambience, price*, etc. They form the key elements of the reviews about which users express their likes or dislikes.

- **Descriptors:** Descriptors are words that occur in the neighborhood of Aspects, and either describe the Aspect, or contain underlying sentiment associated with the Aspect. Examples include *tasty, good, disgusting, expensive*, etc.

The following is a review excerpt from our dataset with the Aspects in bold and the Descriptors in italics:

"Let there be no question: Alexions owns the *best* **cheeseburger** in the region and they have now for decades. The **service** is *flawlessly friendly*, the **food** is *amazing*, and the wings? Oh the wings... but it's still about the cheeseburger. The **atmosphere** is *inviting*.... "

As is evident, the review consists of several key aspects of the restaurant that the user comments on, such as **food**, **service**, and **atmosphere**. The Descriptor words that accompany these Aspect words carry the sentiment of the user with respect to the corresponding Aspect, e.g., the word *inviting* expresses that the atmosphere of the restaurant was perceived positively by the user.

5.4.2 Some Background on Word2Vec

Before discussing the next steps we take for Aspect-Based Sentiment Analysis, we provide a brief introduction to Word2Vec, which is a tool we use for the purpose of our work.

Word2Vec is a deep-learning inspired method that generates distributed representations of words based on the contexts in which they occur. The idea behind this concept lies in the Distributional Hypothesis in Linguistics. This hypothesis is derived from the semantic theory of language use, i.e. words that occur in the same contexts are likely to carry similar meanings [168]. The idea that "a word is characterized by the company it keeps" was popularized by Firth[169]. Traditionally, vector space representations for words were generated by exploring the distribution of the contexts they occurred in. More recently, neural networks are being used for this purpose, as will be subsequently discussed.

Continuous Bag of Words (CBOW) Model and Skip-gram Model: The seminal paper on Word2Vec was written by Mikolov et. al [162]. They proposed the Continuous-Bag-of-Words Model (Figure 5.1) and the Skip-Gram Model (Figure 5.2) for producing vector representations

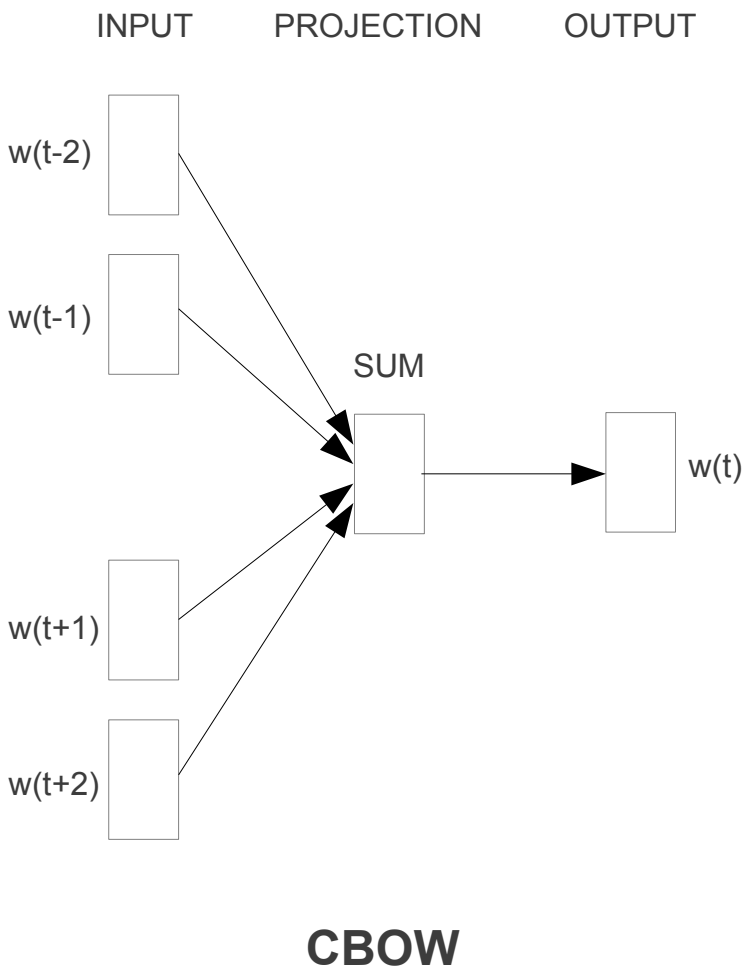


Figure 5.1: Continuous-Bag-of-Words Model Architecture

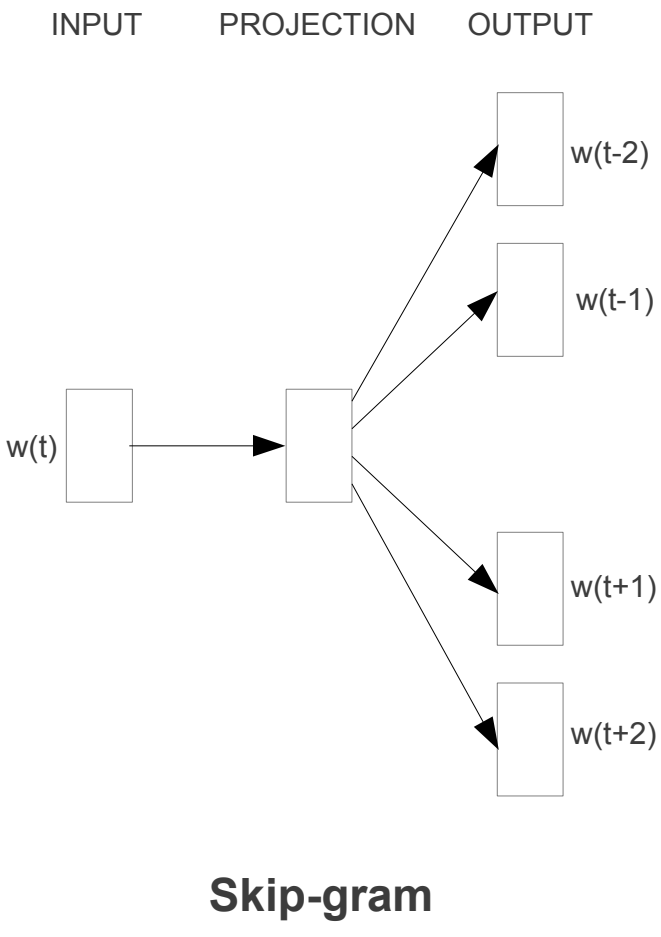


Figure 5.2: Skip Gram Model Architecture

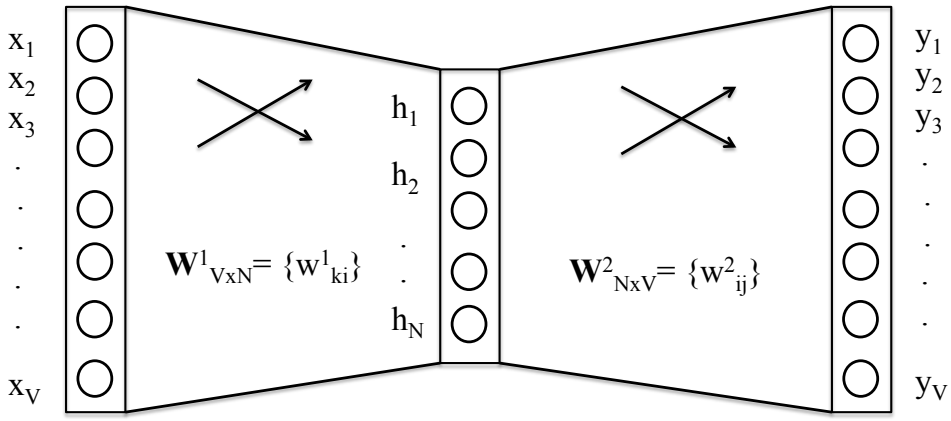


Figure 5.3: Workings of the neural network model for Skip-Gram for only one word in the context

of words in a corpus. The Skip-Gram model (Fig 5.2) learns the word representations by predicting the context of a word, given a word. The context of a word refers to the neighborhood of the given word, i.e. the words that occur before and after the target word. For instance, in Fig 5.2, given the word $w(t)$, the model tries to predict the two words that occur before it and the two words that occur after it. The number of words to be considered is determined by a parameter of the model called *window size*. In Fig 5.2, window size is 2.

Given a sequence of training words $w_1, w_2, ..., w_T$, the objective of the Skip-Gram model is to maximize the following objective function:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c,\neq 0}\log p(w_{t+j}|w_t) \tag{5.1}$$

where c refers to the window size.

The details of the Skip-Gram model can be explained using Figure 5.3, which represents the neural network model assuming that there is only one context word, given a target word. Thus, only one context word is predicted, given the target word. (The same model can be extended for multiple words in the context.) V is the size of the vocabulary and the input $x_1, x_2, ..., x_V$ represents the one-hot encoding of the target word which means that for a given target word,

only one of the nodes of $x_1, x_2, ..., x_V$ will be 1, and the rest will be 0. There are N neurons in the hidden neural network layer that are represented by $h_1, h_2, ..., h_N$. The following softmax function represents the relationship between the input and the neural network layer:

$$h_i = \frac{\exp(\mathbf{x}^T \mathbf{W}^1_{:i})}{\sum_{j=1}^{N} \exp(\mathbf{x}^T \mathbf{W}^1_{:j})} \tag{5.2}$$

$y_1, y_2, ..., y_V$ denotes the one-hot encoding vector of the context word. The probability $p(y_k = 1|x)$ is given by:

$$p(y_k = 1|x) = \frac{\exp(h^T \mathbf{W}^2_{:k})}{\sum_{k=1}^{V} \exp(\mathbf{h}^T \mathbf{W}^2_{:k})} \tag{5.3}$$

The model is trained on all words of the vocabulary. After training the model, $(h_1, h_2, ..., h_N)$ is the vector representation of the target word.

The CBOW model [162] is similar to the Skip-Gram model [162] but tries to maximize the probability of the target word given the neighboring words, as shown in Figure 5.1. We used the Skip-Gram model for our experiments.

5.4.3 Towards Aspect-Based Sentiment Analysis

Our next steps towards obtaining informative features for Aspect-Based Sentiment Analysis are as follows.

1. **Building Subgroups Using Contextually Similar Words:** In user-generated text, a given concept may often be expressed by different word choices by different users, some of which may even be misspellings. We leveraged the Word2Vec model to map all contextually similar words to the same word. Table 5.2 illustrates a few such examples. We then defined **sub-groups** of Aspects and Descriptors, such that words that are contextually similar are placed within the same sub-group. Table 5.3 illustrates some examples of these sub-groups.

Table 5.2: Instances of Aspect and Descriptor Seed Words, their meanings and some of their contextually closest words, computed using cosine similarity. Misspellings and informal language are in bold.

Type of Word	**Seed Word**	**Meaning**	**Contextually Closest Words**
Aspect	*food*	Comments on the food and drinks that were served	foods, meals, meal, pizza, cuisine, sushi, burgers, wine, drink
	ambience	Comments on the general environment and vibe of the place.	**ambiance**, atmosphere, environment, vibe, **decore**, setting, layout, interior
	service	Comments on the behavior of the waiter/waitress/bartender/manager and the service received.	**sevice**,services, relations, **svc**, waitstaff
Descriptor	*delicious*	Expressions of the taste of the food served.	**delish**, **delicous**, delectable, **delcious**, **tastey**, tasty
	dirty	Descriptions of the general cleanliness of the place, the food served, etc.	filthy, unclean, smelly, sticky, stained
	professional	Descriptions of the service received from the waiters or the management.	polite, personable, attentive, courteous, hospitable, efficient, respectful

2. **Construction of Meta-features:** To determine the *sentiment* associated with each Aspect of a restaurant, we proposed the construction of **meta-features**. We defined these as unordered 2-tuples of the form (a_i, d_j) where a_i represents a word from Aspect sub-group i and d_j represents a word from Descriptor sub-group j, such that the words from d_j occur within a neighborhood m of the aspect word a_i. For example, in the sentence *"I didn't enjoy eating here - the ambience sucks"*, considering $m = 1$, $(ambience, sucks)$ represents a meta-feature that captures the negative sentiment associated with the unpleasant ambience of the restaurant. The goal behind constructing meta-features is two-

Table 5.3: A few Aspect and Descriptor sub-groups obtained using contextually similar words. These sub-groups were used to build Meta-Features.

Word Type	**Seed Word**	**Instances of Words in the Subgroup**
Aspect	*ambience*	environment, artwork, decour, atmosphere, atmophere, scenery, openness, decoration, atmostphere, decors, decore, vibe, decorations, furnishings
	portion	quantities, portions, quantity, quanity, helping, value, portion, amount, sizing, serving, size
	food	foods, meals, menu, selection, pizza, burgers
Descriptor	*expensive*	pricy, priciest, overpriced, inflated, astronomical, exorbitant, unjustified, outrageous, steep
	clean	sanitary, tidy, spotless, orderly, immaculately, spotlessly , cleaning, cleaned, cleans, neat, squeaky, hygienic
	delicious	tasty, flavorful, delish, delcious, yummers, homemade, onolicious, mouthwatering, addictive,

fold: (1) they help us in capturing the sentiment associated with the Aspects of the reviewed restaurant, and (2) they transform reviews from a large corpus of millions of words to a small set of rich meta features that makes information extraction and analysis easier.

5.4.4 Verification of Proposed Method: Binary Classification

To complete the task of aspect-based sentiment analysis, we must estimate the sentiment-carrying capacity of the meta-features that we determine. In order to do so, we formulated a binary classification problem using logistic regression with l_2 regularization (to prevent over-

fitting [136]). Each review acted as a data sample, with the class label given by the rating as mentioned in Section 5.3.

In the logistic regression model, $\mathbf{x}_i$ is a data vector of size $k \times 1$ for data sample i, where x_{ij} denotes the frequency of the j^{th} meta-feature in the i^{th} data sample. k is the number of meta-features. y_i is the label of the i^{th} data sample in $\{-1, 1\}$, which is obtained using the numeric ratings as elaborated in Section 5.3.

For the i^{th} sample, the probability that it belongs to the positive class is given by:

$$P(y_i = 1|\mathbf{x}_i, \beta) = \frac{1}{1 + \exp\left(-\beta^T \mathbf{x}_i\right)}, \tag{5.4}$$

where β is a $k \times 1$ coefficient vector. In order to prevent over-fitting [136], we minimize an l_2-regularized logistic loss function to learn β:

$$\begin{aligned} L(\beta) &= -\log\left(\prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \beta)\right) + \lambda\|\beta\|_2^2 \\ &= \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i(\beta^T \mathbf{x}_i)\right)\right) + \lambda\|\beta\|_2^2, \end{aligned} \tag{5.5}$$

where n is the number of samples and λ is the regularization parameter. Thus, given a set of meta-features $\mathbf{x}$ and a set of known outputs y in the training data, the logistic regression model learns the parameter β that determines the relationship between $\mathbf{x}$ and y. Once the model has been learned, it can then be used to predict the labels of the test data, given their meta-features $\mathbf{x}$.

5.5 Implementation Details

In this section, we discuss in detail the implementation of each step of our proposed methodology. As mentioned in Section 5.3, only the training data was used for all the steps of

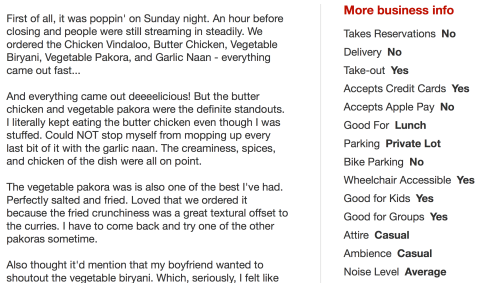


Figure 5.4: Snapshot of a restaurant review page on Yelp.com [5]

the pipeline up to the Classification step (Sections 5.5.1-5.5.3). The test data was used in Section 5.5.4. We used the numerical ratings only for the classification step, and used the textual data for the initial steps.

5.5.1 Training Word2Vec on Review Data

We used the Python package *gensim* [170] for training Word2Vec, which implements the Skip-Gram model [162]. The input to the model is an ordered sequence of words. The only data pre-processing we performed is to convert the review text into lowercase, to deal with case-insensitivity. Each sentence of a review was tokenized into a sequence of words using Python's NLTK package [171] and fed into the model. There are 3 primary parameters for the model training, namely the word vector dimensions N, the window size w and the minimum frequency count f. N dictates the size of the word embeddings, w determines the size of the neighborhood given a target word, and f represents the minimum number of times a word has to appear in the vocabulary to be a part of model training. We use $N = 150$, $w = 5$ and $f = 20$ in our experiments. After training the model, we now have numerical embeddings of size N for each word in the vocabulary that occurs at least f times.

5.5.2 Extracting Key Drivers of Sentiment

To extract Aspects and Descriptors from the reviews, we used the following method:

1. **Determining Aspects and Descriptors**

   ***Aspects*:** We first picked a few seed Aspect words by consulting the Yelp website [5]. Yelp pages containing the reviews of restaurants usually contain a series of features on the right side under "More business info" (Figure 5.4). These contain information on whether the restaurant delivers food, accepts credit cards, has parking, is good for kids, etc. By picking keywords from these features, we obtained 22 seed words for Aspects, namely *attire, ambience, food, reservations, delivery, payment, cost, portions, taste, service, parking, preparation, celebration, lunch, kids, family, tv, location, clientele, wifi, website, cleanliness*. A few of these Aspect seed words are explained in Table 5.2. The words were chosen such that they span the aspects on which restaurants would be reviewed by users.

   ***Descriptors*:** We explored the neighborhood of Aspect seed words in the training data to obtain Descriptor seed words. For each Aspect seed word in the training data, we extracted the co-occurring words (excluding stopwords) from a 5-window neighborhood of the seed word. We then obtained the overall frequency of occurrence of these neighboring words. The 100 most frequently occurring words are manually examined and 21 of them were labeled as Descriptor seed words.

2. **Obtaining Sub-groups of Words:** For each of the Aspect and Descriptor seed words, we determined their contextually closest words by using cosine similarity on their word embeddings. We used a threshold of 0.5 and select words whose cosine similarity is larger than the threshold. We found the quality of the closest words to drop below that threshold, for most words. Table 5.2 contains a few instances of the closest words obtained using Word2Vec. Further, to ensure that each sub-group captured a unique concept and is different from other sub-groups, we unified any pair of sub-groups if the majority of words in either of them are the same. This resulted in merging a couple of Descriptor

sub-groups. Thus, we obtained 22 Aspect sub-groups and 20 Descriptor sub-groups.

5.5.3 Meta-Feature Construction

To extract meta-features from a data sample, we located Aspect words (collected in Section 5.5.2) in all sentences of the sample. For every Aspect word, we located Descriptor words within a neighborhood of 5 words within that sentence. We disregarded stopwords during this process.

For example, in the following data sample:

"I'm giving 4 stars mostly because of the beer....large selection & decent prices. The food is pretty good, but nothing to rave about. The menu has a good variety, and everything I've tried has been good. Portions are large.",

(portions, large) would be one such unordered 2-tuple since *portions* is an Aspect and *large* is a Descriptor. Suppose *portions* belongs to Aspect sub-group 1 and *large* belongs to Descriptor sub-group 5. Then this meta-feature would be indexed $(1, 5)$. If, from a different sentence, we obtain the tuple *(serving, big)*, this meta-feature would also be indexed by $(1, 5)$, since *serving* and *portions* belong to the same Aspect sub-group, and *big* and *large* belong to the same Descriptor sub-group. We have 438 meta-features in all.

5.5.4 Binary Classification

Using the meta-features that we construct, we now look for the frequency of occurrence of these meta-features across the training and test datasets, to build our data matrices. There are 438 meta-features, 509,902 training samples and 101,794 testing samples. The label distribution across both matrices is 62.82% positive and 37.18% negative.

5.6 Results

In this section, we outline the POS tagging based method we compare with, and present the experimental results obtained using our proposed method. Further, we perform comparative analysis on restaurants, and present those results as well.

5.6.1 POS Tagging as a Comparative Baseline

Parts-of-Speech (POS) tagging being a very popular method employed for Aspect-Based Sentiment Analysis [48, 49, 73], we decided to compare our proposed method with a similar pipeline generated using POS tagging. To ensure a fair comparison, we simply replaced the use of Word2Vec in our proposed scheme with that of POS tagging and kept the rest of the pipeline the same. Thus, we still constructed meta-features for the comparison, except that we used POS tagging to obtain them. This would enable us to effectively evaluate the necessity of Word2Vec.

Similar to the Word2Vec training approach we adopt, we converted the reviews to lowercase, and tagged our training data using a very popular POS tagger, the Stanford POS Tagger [172]. Since Aspects, by definition, are most likely to be nouns, we pulled out the "NN" (nouns) and "NNP" (noun phrases) tagged words from the data. This is similar to the approach taken in [48] for aspect extraction. Further, since Descriptors are most likely to be adjectives, we then looked for the presence of "JJ" (adjective) tags within a 5-window neighborhood of nouns. Stopwords are ignored in this process, in the same vein as our proposed method. Each (noun, adjective) pair constitutes a meta-feature, and we collected those that occurred at least 20 times. There are 13,487 meta-features in all. To compare the methods, we then trained the same classifier (5.5) using these meta-features.

Table 5.4: Classifier Metrics using l_2-regularized Logistic Regression

Method	**Overall Accuracy (%)**	**Precision**	**Recall**	**Specificity**	**AUC**
Proposed Method	**79.43**	**0.838**	**0.839**	**0.716**	**0.777**
POS Tagging Based Method	67.93	0.792	0.732	0.528	0.63

5.6.2 Method Validation and Comparison

To quantitatively validate our proposed method, we report the usual classification metrics [151] in Table 5.4. The metrics reported are accuracy, precision, recall, specificity and AUC. The best results are obtained with the regularization coefficient 0.0001. In order to compare with the POS Tagging based method, we report the same metrics for that method as well. As can be observed, our method performed better than the POS-based method, for all classification metrics. For instance, the overall accuracy obtained using our method was 79.43%, whereas that obtained by the POS tagging based method was 67.93%. The main shortcoming of the latter seemed to be in capturing negative sentiment, since the specificity achieved was 0.528.

Using the sign and magnitude of each component of the feature weight vector β (6.2) learned by the classifier, we obtained an explicit sentiment weight for each meta-feature. In Table 5.5, we demonstrate a few meta-features that have the highest positive and negative weights, and were deemed the most discriminative by the classifier. As expected, *(food, delicious)*, *(service, speedy)*, *(price, reasonable)* are all instances of meta features that express positive sentiment, while *(cleanliness, dirty)*, *(food, disgusting)*, *(taste, bland)* convey negative sentiment. Thus, we find positive feature weights to correlate with positive sentiment while

Table 5.5: Instances of the most positive and negative meta features uncovered from the feature weights during classifier training, using our proposed method as well as using Parts-of-Speech tags. The sign and magnitude of the feature weight vector was utilised in obtaining them.

Method	**Sentiment**	**Meta Features**
Proposed Method	*Positive*	**(parking, efficient), (attire, classy), (food, delicious), (reservations,speedy), (price, reasonable), (preparation, clean), (delivery, delicious), (service, speedy), (portions, generous), (family, accommodating)**
	Negative	**(service, disgusting), (delivery, negligent), (food, disgusting), (taste, mediocre), (preparation, disgusting), (cleanliness, disgusting), (wifi, disgusting), (food, dirty), (service, unhelpful), (taste, bland)**
POS Tagging	*Positive*	(wine,wonderful), (pizza,wonderful), (bistro, french), (world, top), (coffee, wonderful), (cuisine, great), (tap, great), (pho, phoenix), (diner, welcome)
	Negative	(pizza, frozen), (bread, old), (salad, frozen), (seafood, old), (cheese, frozen), (steak, frozen), (buffet, golden), (chicken, fine), (wings, frozen), (sub, mexican)

Figure 5.5: Word Cloud Representing the Most Popularly Used Positive Meta Features. The larger the size of a word, the greater its frequency of occurrence.

negative feature weights correlate with negative sentiment. Making this distinction enables us to do further detailed analyses on users' likes or dislikes about restaurants.

We also compared the meta-features discovered using the POS tagging based method in Table 5.5. It is interesting to observe that amongst the most highly weighted features, the variety of the Aspects captured is very less. Most of the meta-features it discovers are on similar Aspects, e.g. *(wine,wonderful), (coffee, wonderful), (pizza, frozen)* are all references to the food and drinks served in the restaurants. In contrast, our method is able to discover a larger variety of review Aspects, even though they may not be correlated, e.g. *(food, delicious), (attire, classy), (cleanliness, disgusting)*. This allows for a wider coverage of consumer sentiments on a variety of subjects. Also, it is to be noted that the meta-features obtained using the POS tagging method are representative tuples of the meta features of our method.

Thus the classifier helps in identification of meaningful, sentiment-carrying meta-features, enabling us to understand consumer sentiment at a more granular level.

Figures 5.5 and 5.6 represent word clouds we constructed using the occurrence frequency

Figure 5.6: Word Cloud Representing the Most Popularly Used Negative Meta Features. The larger the size of a word, the greater its frequency of occurrence.

of the positive and negative meta-features. The sizes of the text represent the frequency of occurrence of the meta-features. We use an online tool: WordItOut[173] to build the word clouds. As is evident, meta-features such as *(food, amazing), (service, amazing), (ambience, amazing)* are the most frequently used phrases by people when they express positive sentiment w.r.t restaurants. Similarly, phrases such as *(food, slow), (food, disgusting), (food, mediocre)* reflect the most popular reasons for users to dislike a restaurant.

5.6.3 Coverage using Meta-Features

Using the Word2Vec model to construct meta-features has enabled us to capture contextually similar words that may be literally different but semantically similar. This has enabled the coverage of a larger fraction of the data than would have otherwise been possible. For instance, simply looking for the presence of a tuple of Aspect and Descriptor seed words, such as *(food, delicious)* would cover a smaller portion of the entire dataset than looking for the cor-

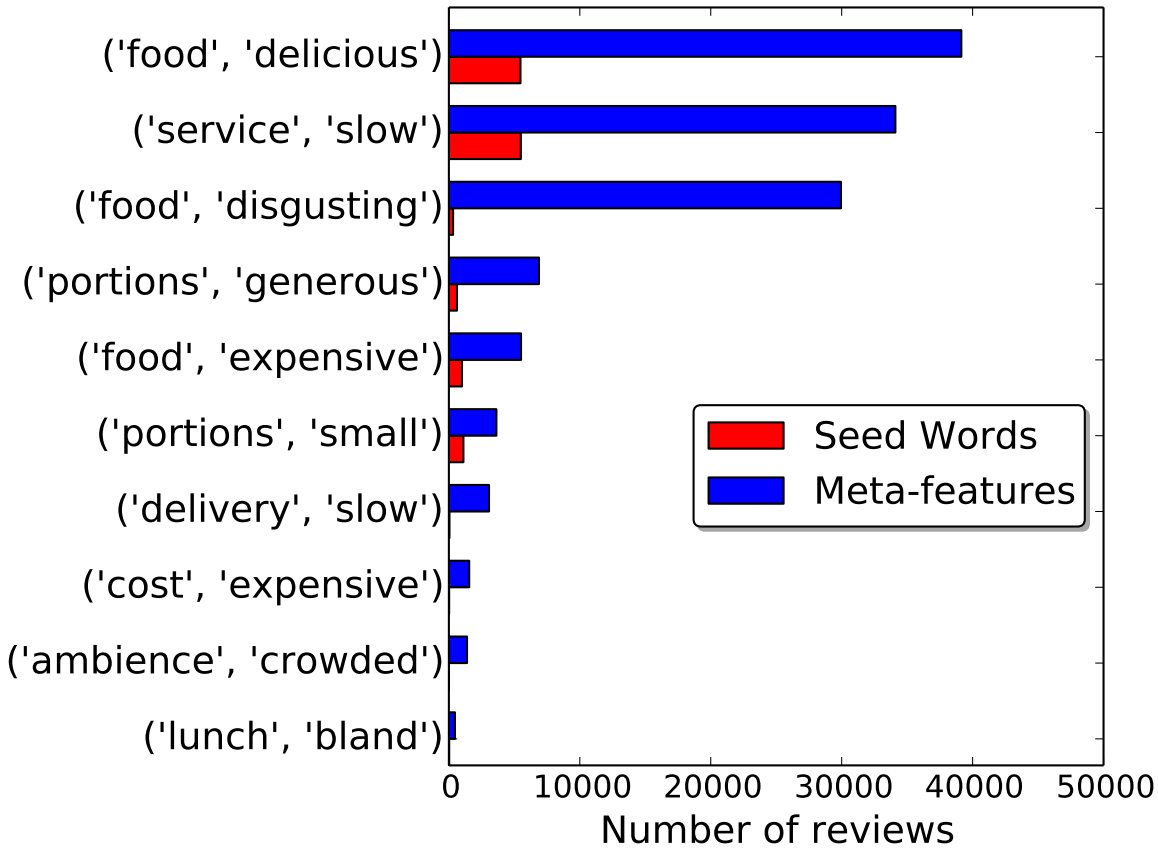


Figure 5.7: Difference in Coverage Obtained by using Meta-Features vs Tuples of Seed Words

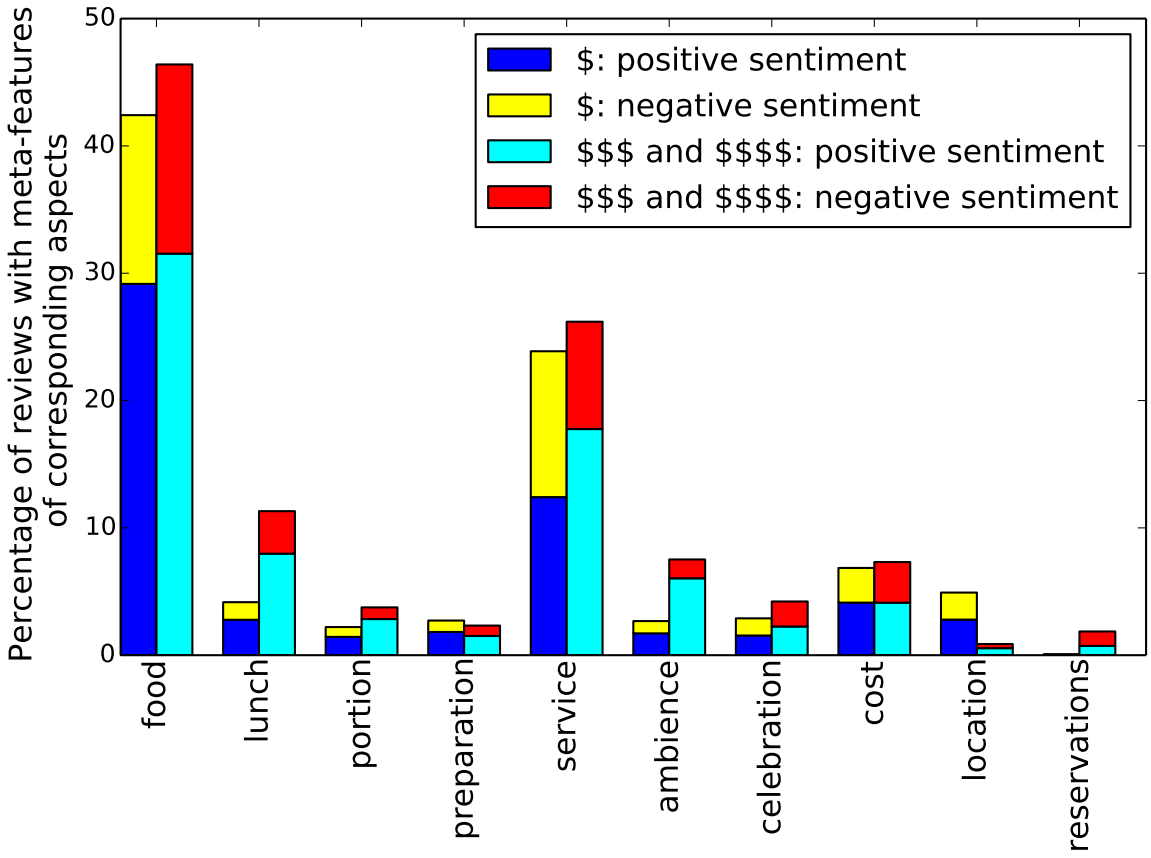


Figure 5.8: Comparative Analysis of High-end and Low-end Restaurants. For each Aspect on the x-axis, the y-axis contains the percentage of reviews of the two kinds of restaurants that mention the corresponding meta-features. Differences in color denote different sentiments.

responding meta-feature, which captures variations in words, mis-spellings, etc. In Fig 5.7, we plot the occurrences of a few examples of seed word tokens and those of their corresponding meta-features, to illustrate the increase in coverage we achieve using the meta-features.

5.6.4 Comparative Analyses on Restaurants

The sentiment-carrying capacity of meta-features allowed us to perform interesting comparative studies on restaurants, at a granular level. One such study is to compare, on an Aspect-level, which Aspects people review the most and what opinions they express on them when re-

viewing high-end restaurants vs inexpensive ones. Yelp.com [5] carries information regarding the food prices of restaurants using "$" signs next to the name of the restaurant. "$" implies a cheaper eating place, whereas "$$$" or "$$$$" imply an expensive place. We harness this information when collecting data for the following analysis.

We first extracted the 50 most reviewed restaurants present in our dataset, and then determine the low-end and high-end restaurants (as described above) out of those by consulting Yelp.com [5]. Out of these restaurants, we have 11 low-end ones with a total of 33,093 reviews and 9 high-end ones with 23,512 reviews. The rest were "$$" restaurants that we do not consider for the comparison since we were interested in analysing restaurants that lie on two ends of the price spectrum. We then obtained the positive and negative meta-features present in both sets of reviews, Aspect-wise, and plotted the ten most popular Aspects and the corresponding aggregated sentiment in Fig 5.8.

As expected, ***food*** is by far the single most reviewed Aspect across both kinds of restaurants. Next comes ***service***, and we can observe that the high-end restaurants are slightly more positively viewed in terms of this Aspect. Another interesting but expected observation is that ***ambience*** is mentioned more often in the reviews of the high-end restaurants and is associated with a higher positive sentiment as well. This reflects that clientele of expensive restaurants take into consideration the ambience, while such is usually not the case in inexpensive places. Further, cost being the distinguishing factor for the two sets of restaurants here, it is interesting to observe that for the Aspect ***cost***, the number of mentions as well as the proportion of positive and negative sentiment are comparable. The possible explanation for this is that cost is judged on the basis of the food/service received, and not just on the amount of money spent. On closer examination of the reviews, it seems that users are well-aware of the prices of the restaurants they choose, and more often than not are satisfied with the value for money they get. For example, the first two review excerpts are from low-end restaurants and the next two are from expensive ones:

"Was in LV for the wknd was looking for a place to have dinner that wouldn't cost $$$$ this place was it."

"...In a city (especially the Strip) loaded with overpriced, overcooked and unremarkable food - the Burger Bar is a fabulous find."

"...dinner cost about $230 with tip, which wasn't too bad..."

"...The reviews hating on the cost of bread are out of control. Did you go to Bouchon for a good deal? I hope not. It's expensive. We spent $100 on brunch and honestly thought we got out of there for a steal..."

Thus, a customer of a restaurant, whether expensive or cheaper, is more likely to leave a positive review if she enjoys her overall eating experience, irrespective of the amount of money she spent. As far as other Aspects are concerned, ***reservations*** are discussed w.r.t. the high-end restaurants since these are more likely to require or even offer reservations. Further, ***location*** is discussed more for the inexpensive restaurants since users may not want to go out of their way to eat at these places.

5.7 Additional Experiments

We tested a similar pipeline on a different dataset, namely, a dataset of digital camera reviews from Amazon. The results obtained on that dataset are provided in Appendix A.

5.8 Conclusions

In this work we demonstrate a method for representing a large corpus of user-generated restaurant reviews by a feature set that captures the *what*, *how*, and *why* of ratings: what aspects customers care most about in a restaurant, how they feel about those aspects, and why. By using contextual embeddings of words we are able to identify and aggregate textual variations with

similar meaning, and reduce feature space from 100M tokens to 438 meta-features, achieving strong statistical power while maintaining high coverage of the original corpus. We show that these meta-features have strong predictive power of sentiment, and hence can be used as a way to automatically extract aspect-level feedback from customers automatically and at scale. Our method also enables us to perform comparisons between different kinds of restaurants by analyzing aspect-level sentiment. The method can be extended to other types of reviews as well, as we have shown in Appendix A.

Chapter 6

Automatic Detection of Age and Conversational Topics of Twitter Users using Distributed Representation of Words

6.1 Introduction

In today's day and age, a large section of the human population uses some form of online social network. For instance, 62% of the entire adult population in the U.S. uses Facebook [174]. With the continuing increase in use and popularity of online social networks today, researchers have a plethora of opportunities to analyze online user behavior. Inferring user interests and attributes from their online posts is an important area of research, especially since it helps in user profiling, as discussed in Chapter 2

It is a well-known fact that people's behavior changes as they age. The conversational topics they engage in, and the way in which they express themselves, change as they grow older.

In this work, our goal is to determine the age of users from their self-generated tweets, by identifying the different topics of conversation they engage in. User age detection is important from a number of perspectives. First, not all online websites and social media platforms ask for age confirmation when signing up, and even if they do, they are relying on the users' transparency and honesty. Further, most OSNs have a low age limit for users to sign up on their networks. For instance, Facebook, Twitter, Instagram, Pinterest, Snapchat require account holders to be at least 13 years of age to sign up [175, 176]. Our technique for age detection provides an approach relying on a person's behavior on the social network. This is crucial, especially to prevent age-restricted content from being displayed to younger age groups. For instance, one article [175] reported that 10 and 12-year-olds in the U.K. have signed up on various popular OSNs and became targets for online hate. Moreover, detecting the age of users helps in user profiling (Chapter 2.1.4), which is advantageous for recommendation, personalization, and advertising.

We collect English tweets of randomly selected Twitter users over a period of 10 months. We divide the users into two groups - below 21 years of age, and 21 and above. This age cutoff was arbitrarily chosen since we expect a shift in user behavior as they grow out of their teens into an older age. Other age cutoffs such as the age of legal consent could also be selected, however, these ages vary at a state level across the U.S. Thus, we chose the legal age for alcohol consumption (21 years) which remains universal across all states of the U.S. The method we propose can be applied to any age cutoff, depending on the purpose at hand. We obtain the age labels by looking for mentions of actual age in the profiles of users, using text annotators. The method we propose determines the age of users by identifying the topics of conversation they engage in.

6.2 Related Work

Understanding latent attributes of users such as their age, gender etc. from their OSN posts is an important step in the process of user profiling. In Nguyen et. al's work [85] the relationship between the age of users and the language they use on Twitter is studied. They address a similar problem to ours, in detecting ages of users using their publicly available tweets. They perform age detection in three ways: by classifying users into age categories, by life stages, and by predicting their exact age. They find unigrams extracted from tweets to be the best predictors of age and use them as features in a classification scheme to predict age. Their task of classifying users into age categories, namely, below 20 years of age, 20-40 and above 40 years, yields the best results.

The above work differs from our approach in a number of ways. First, their dataset consists of Dutch tweets as opposed to English, and contains a much smaller set of users. For the age categorization task, they have 3110 users in all, as opposed to $> 70,000$ users in our case. One difference in approach is the procedure used for obtaining ground truth labels. They employed human annotators to obtain the age labels by examining the actual tweets or the profile of the user concerned, as well as their accounts on external platforms such as Facebook and LinkedIn. In contrast, we use automatic text annotators and only look for explicit mentions of user ages in their Twitter profiles. In practice, it is infeasible and expensive to label $> 70,000$ users using human annotators. Second, their age detection problem involves categorizing users into 3 different age categories, into various life stages and predicting their exact age, while our task involves classifying users into < 21 years of age and $>= 21$ years of age. Third, their method employs a generic bag-of-words approach, i.e. the use of unigrams as features in a multiclass logistic regression model, while we construct features using distributed representation of words and clustering and use it for our classification scheme. Fourth, their method does not capture the different topics of discussion amongst the age groups, which our method automatically

achieves. Fifth, they disregard the use of usernames such as *@name* in the tweets, whereas we consider them important especially since users tend to mention celebrities or popular Twitter handles, or retweet them, e.g. *@justinbieber, @adele, @thatbucketlist* to a large extent in their tweets. However, since this work addresses a similar problem to ours, we compare with their method as described in subsequent sections.

The difficulty associated with age detection from tweets has been addressed in [177], in which the authors demonstrate that users don't always express their biological age in their tweets, owing to a difference in the identities they hold on OSN platforms and their actual biological identities. This makes detecting these latent attributes from OSN posts a difficult task.

6.3 Dataset and Challenges

In this section, we elaborate on the process implemented for collecting data and obtaining ground truth labels. Further, we also highlight the challenges faced in addressing the problem at hand.

6.3.1 Data Collection

Tweets were collected using Twitter's Decahose stream (which was provided to IBM Research) over 10 months. 182 million users and 11 billion tweets were collected in all. The profile descriptions of users were collected as well. After spammers were removed, we were left with 181 million users. In order to verify our proposed method, we required ground truth labels, for which we extracted explicit age mentions from user profile descriptions. Here are a couple of examples of user profiles where they explicitly mention their age:

Twitter Profile 1: "*More watermelon, less crack. worst COD player in history. 17 years old. Instagram: beardedconfusion.*"

Twitter Profile 2: "*like: music,films * hobby: drawing,polymer clay * twenty two years old*"

Text annotators were used to detect age mentions. We ensured that only users with their actual age explicitly mentioned were picked as part of our dataset. There were 1,320,309 such users. Our experiments were conducted on a subset of these users, randomly picked. Thereafter, we extracted users that had at least 20 tweets in the dataset such that we had enough tweets to detect their age from. This is similar to the approach used by the authors in [85]. We are thus left with 72,003 users and 7,039,643 tweets in all.

Since our problem involves identifying users that are old enough to view mature content, we divide our data into 2 classes: users who are below 21 years of age and users who are 21 and above. Owing to the inherent population bias in Twitter, the younger age group is more prevalent in our data. The distribution is 75.15% below 21, and 24.85 % for 21 and above. Since we use a classification scheme as part of our method, we divide the data into training and test sets for the sake of our experiments. Using stratified sampling to preserve the class distribution, we divide the set of 72,003 users into 75% training and 25% test data sets. Thus, we have 54,010 users in the training set and 17,993 users in the test set.

6.3.2 Challenges Faced

Many of the challenges in working with Twitter data have been discussed in Chapter 1.4. Further, our dataset consisted of generic tweets on a myriad of topics that people chose to talk about on Twitter, rather than on a single topic. Thus, this posed a problem for focused signal extraction. Further, age detection is a hard problem in itself. In [177], the authors discuss the difficulties associated with detecting age and gender from tweets. The main difficulty, according to [177], is that there may be a vast difference in the social and biological identities of the users on platforms such as Twitter, owing to which their tweeting behavior is often very different from what would be expected, given their age.

Here are a few examples of tweets of different users in our dataset:

Tweet 1: *"If you could pick an eye color what would you choose? Color? I wish they are shaped like cookies."*

Tweet 2: *"@AwkwardWenna I deserve all the awards"*

Tweet 3: *"RT @billboard: Why @BritneySpears and @IggyAzalea's new single could be mutually beneficial: http://t.co/GzfNBsU69r http://t.co/w0VozoIV5c"*

Tweet 4: *"Soundtrack 2 My Life"*

Tweet 5: *"RT @NiallOfficial: .@TheXFactor is back tonight , and mommy Roch @RochelleHumes is on xtra factor at 9:30 !"*

Tweets 1, 2, 4 and 5 belong to users who are below 21 years of age while Tweet 3 belongs to the older age group. As may be observed, the tweets are very similar to each other in the sense that they all use a very similar language. This makes it challenging to identify the age of the user tweeting.

6.4 Identifying Topics of Conversation

It is expected that users of different age groups would indulge in different topics of conversation. The key idea of the method we propose is to aggregate the tweets of users and identify these topics in an automated manner. These conversational topics, once identified, were then used as features in order to predict the age of users.

6.4.1 Distributed Representation of Words

As discussed in Section 5.4.2, Word2Vec [162, 163] is a method for computing distributed representation of words, while maintaining their contextual similarity. Since it is expected that words of similar topics would be used in a similar context to one another, we propose the use of Word2Vec in order to uncover these conversational topics in an automated fashion.

We trained Word2Vec on the tweets in our training data (details of which will be discussed subsequently), to obtain word embeddings for the words in the dataset. These embeddings are used in the next steps.

6.4.2 Clustering

To determine the topics of conversation, we propose the use of a clustering scheme over the word embeddings, to group similar word embeddings together. k-means [178, 179] is a popularly used clustering algorithm that aims to partition p samples into k clusters, where each cluster is represented by a *mean* point, or centroid, and a sample is assigned to the cluster with the closest centroid.

Given a set of observations $(\mathbf{h_1}, \mathbf{h_2}, ...\mathbf{h_p})$, where $\mathbf{h_i} \in \Re^m$, k-means minimizes the following objective function:

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{h \in C_i} \|h - \mu_i\|^2 \tag{6.1}$$

where C_i is the ith cluster and μ_i is its centroid.

We clustered the word embeddings such that similar words are clustered together in the same group, thereby making topic detection easier.

6.5 Age Detection

Using the age labels obtained from the user profile as ground truth labels, we propose a binary classification scheme to estimate the age-detection capacity of the cluster-features obtained in Section 6.4.2. We propose the use of logistic regression with l_2 regularization (to prevent over-fitting [136]) for this purpose. In the logistic regression model, $\mathbf{x}_i$ is a data vector of size $k \times 1$ for data sample i, where x_{ij} denotes the frequency of the j^{th} cluster-feature in the i^{th} data sample. k is the number of cluster-features, and y_i is the label of the i^{th} data sample in

$\{-1, 1\}$, which is obtained using the age labels as elaborated in Section 6.3.1.

For the i^{th} sample, the probability that it belongs to the positive (age ≥ 21) class is given by:

$$P(y_i = 1|\mathbf{x}_i, \beta) = \frac{1}{1 + \exp\left(-\beta^T \mathbf{x}_i\right)}, \tag{6.2}$$

where β is a $k \times 1$ coefficient vector.

To prevent over-fitting [136], we minimize an l_2-regularized logistic loss function to learn β:

$$\begin{aligned} L(\beta) &= -\log\left(\prod_{i=1}^{n} P(y_i|\mathbf{x}_i, \beta)\right) + \lambda\|\beta\|_2^2 \\ &= \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i(\beta^T \mathbf{x}_i)\right)\right) + \lambda\|\beta\|_2^2, \end{aligned} \tag{6.3}$$

where n is the number of samples and λ is the regularization parameter. Thus, given a set of features $\mathbf{x}$ and a set of known outputs y in the training data, the logistic regression model learns the parameter β that determines the relationship between $\mathbf{x}$ and y. Once the model has been learned, it can then be used to predict the labels of the test data, given their features $\mathbf{x}$.

6.6 Implementation

In this section, we elaborate on the details of implementing the proposed method. Only the training data was used for all the steps of the pipeline up to the Classification step (Sections 6.6.2-6.6.4). The test data was used in Section 6.6.5. We use the age labels only for the classification step, and use the textual data for the initial steps.

6.6.1 Data pre-processing

For data pre-processing, we converted all tweets into lowercase to preserve uniformity. We also removed URLs since they do not add to our feature extraction process. No other pre-processing steps were performed for the method.

6.6.2 Training the Word2Vec Model

The first step is to obtain word embeddings by training Word2Vec [162, 163] on the training data. We use the Python package *gensim* [170] for training Word2Vec, which implements the Skip-Gram model [162]. The input to the model is an ordered sequence of words. We tokenize each tweet in our dataset using Python's NLTK package [171] and feed it into the model. There are 3 primary parameters for the model training, namely the word vector dimensions N, the window size w and the minimum frequency count f. N dictates the size of the word embeddings, w determines the size of the neighborhood given a target word, and f represents the minimum number of times a word has to appear in the vocabulary to be a part of model training. We use $N = 300$, $w = 5$ and $f = 10$ in our experiments. After training the model, we now have numerical embeddings of size N for each word in the vocabulary that occurs at least f times.

6.6.3 Unigram selection

To obtain meaningful clusters of words representing topics of conversation, we first select unigrams that would be most informative. The following are the steps undertaken for the process:

1. **Frequency of occurrence and Stop words:** We first select unigrams based on frequency, such that we do not pick up words that have been very rarely used. We pick

the unigrams that have occurred at least 20 times in the dataset. This is similar to the approach adopted in [85]. When selecting unigrams, we ensure that we do not include stop words. We use NLTK's [171] stopword list for the purpose.

2. **Odds ratio:** Since the primary goal of the method is to detect age, we aim to pick unigrams that would be the most predictive for the process. Odds ratio [180] is a popularly used method in statistics to determine the association between two properties. We use this measure to determine the association between the presence of unigrams and the class labels in the following manner. We computed the log odds of each unigram selected above, by the following measure:

$$\text{Probability of unigram } w \text{ occurring in class 0, } prob_0 = max\left(\frac{count_0}{|class_0|}, 0.0001\right) \quad (6.4)$$

$$\text{Probability of unigram } w \text{ occurring in class 1, } prob_1 = max\left(\frac{count_1}{|class_1|}, 0.0001\right) \quad (6.5)$$

where $count_0$ and $count_1$ are the number of users in class 0 and class 1 (of the training data), respectively, that have used unigram w. Multiple uses by a single user are only accounted for once. $|class_0|$ and $|class_1|$ represent the class sizes in the training data. Thus, the log odds is computed by:

$$\text{log odds of } w_i = \log\left(\frac{prob_0}{prob_1}\right) \quad (6.6)$$

A unigram with a similar frequency of occurrence in both the classes would not have a very strong predictive power. The log odds of such a unigram would be close to 0. Thus, we picked unigrams whose log odds are either much larger than, or much smaller than 0.

Keeping such unigrams might add noise to the clustering procedure, thereby leading to the detection of noisy topics of conversation. Thus we picked those whose log odds were either greater than 0.01 or less than -0.01. This enabled us to select 72,287 unigrams from 5,152,540 unigrams present in the dataset.

6.6.4 Clustering

After obtaining the word embeddings of the unigrams selected in Section 6.6.3, we cluster them using k-means [178, 179]. On exploring the clusters obtained, we detect the presence of distinct topics of conversation. Table 6.2 illustrates some of these topics corresponding to the word clusters. We experiment with a large range of *k*, the number of clusters, details of which are provided in Section 6.7.1. These clusters are used as features in the next step.

6.6.5 Classification

Using the word clusters as features, we constructed data matrices from the training and test data, where each user is a data sample. For a user i, the frequency of use of words from a word cluster j is the corresponding value in the data matrix X_{ij}. We trained the regularized logistic regression model (6.3) using the training data and then test on the test data to verify our method.

6.7 Experimental Results

In this section, we present the results obtained using our method, and the methods we compare it with.

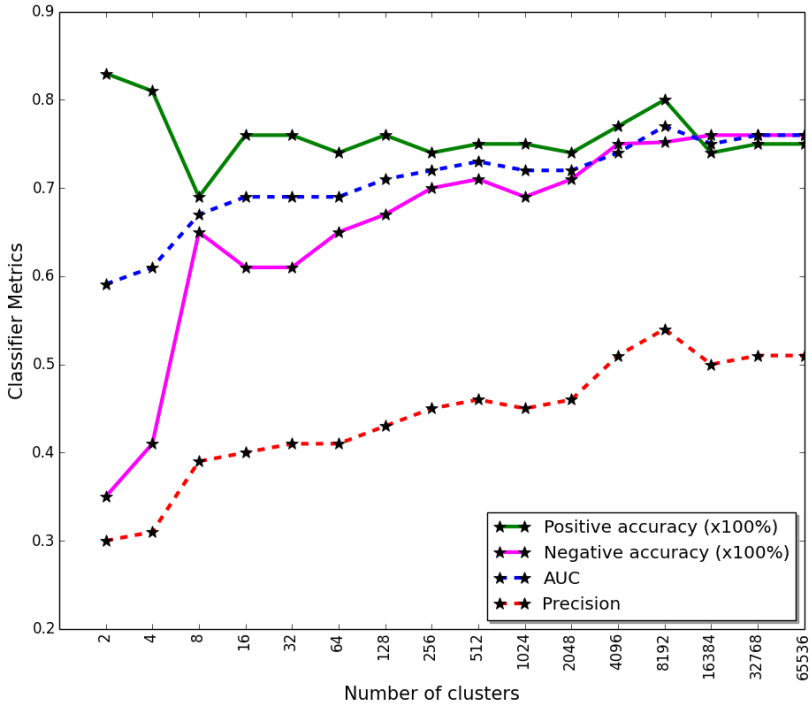


Figure 6.1: Varying k, the number of clusters, over a large range. Results report the classifier metrics using clusters of different sizes.

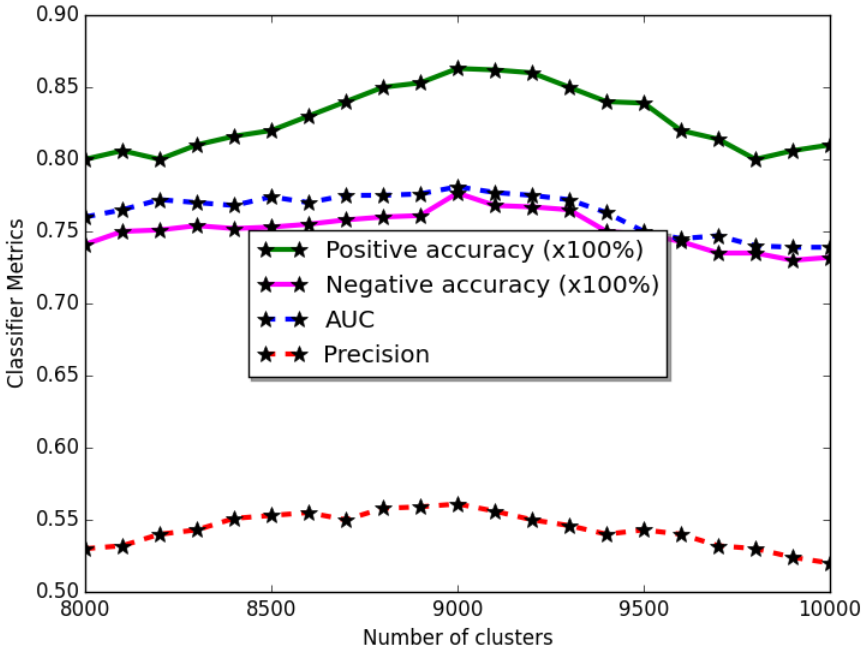


Figure 6.2: Varying k over a smaller range of values, over 8000-10000 in steps of 100.

6.7.1 Best Value of k

We varied the number of clusters, k, over a large range of values from 2^1 to 2^{16}. For each of these values of k, we constructed training and test data matrices, trained the logistic regression model (6.3), and computed the usual classification metric values [151] on the test data. The accuracies of both classes, the precision and the AUC for every value of k are illustrated in Figure 6.1. The best results appear in the neighborhood of 2^{13}, thus in order to determine the best value of k, we varied k over a smaller range of values from 8000 to 10,000, in steps of 100. These results are reported in Figure 6.2. As can be observed, the best results are obtained at $k = 9000$, and are reported in Table 6.1. Overall metric values in this range ($k = 8000$ to $10,000$) are consistent, e.g. mean positive accuracy $= 82.8\%$ with a variance of 0.000468. Similarly, avg AUC $= 0.7634$, with variance of 0.000188.

6.7.2 Comparisons

As mentioned in Section 6.2, we compare with the procedure used by Ngyuyen et. al [85]. In this work, the authors use unigrams that occur at least 10 times in the dataset comprised of users that have at least 20 tweets. Using these selected unigrams as features in a regularized logistic regression model, we varied the regularization parameter λ and report the best results in Table 6.1. As can be observed, our proposed method outperforms this method across all classifier metrics, especially for the metrics of positive accuracy and precision.

To demonstrate the importance of the unigram selection process we adopt (Section 6.6.3), we conduct an experiment in which the logistic regression model was trained using *all* distinct unigrams in the dataset as features, with no frequency-based selection. The results are reported in the 2nd row of Table 6.1. As is evident, the results obtained using our proposed method are better.

Further, in order to compare the Odds Ratio method for unigram selection against other methods of feature selection [181], we conduct experiments using Chi-squared (χ^2) test and Mutual Information. The χ^2 test is a statistical hypothesis test that we use to measure the lack of independence between a unigram and a class label. We use scikit-learn's [148] implementation of the test to obtain χ^2 values for each of the frequency-selected unigrams. Thereafter, we picked 70,000 unigrams with the highest χ^2 values and performed clustering on those unigrams using their corresponding word embeddings. We ran experiments over a large range of *k* (2^1 - 2^{16}), as we did for our method. The best classification results using these clusters are reported in Table 6.1. Similarly, we compute the Mutual Information between each frequency-selected unigram and a class label, and select the unigrams with the 70,000 highest values. Results obtained using these selected unigrams are also reported in Table 6.1. It appears that the odds ratio based unigram selection performs better than other feature selection metrics for this problem.

Table 6.1: Classification Results using the proposed method as well as for some competing methods. The results of the proposed method are in bold.

Method	**Positive Accuracy (%)**	**Negative Accuracy (%)**	**Precision**	**AUC**
Proposed method	**86.3**	**77.66**	**0.561**	**0.78**
All unigrams	74.97	77.52	0.524	0.762
Nguyen et. al [85]	72.95	76.72	0.508	0.748
Unigrams selected by χ^2 + kmeans	73.61	73.28	0.477	0.734
Unigrams selected by mutual information + kmeans	74.5	74.86	0.495	0.747

6.7.3 Detecting Topics of Conversation

Using the sign and magnitude of each component of the feature weight vector β (6.3) learned by the classifier, we obtained an explicit sentiment weight for each cluster-feature. In Table 6.2, we demonstrate a few cluster-features that have the highest positive and negative weights, and were deemed the most discriminative by the classifier. On inspection of the cluster words, we were able to find the topics they were associated with. For instance, the food-related cluster that was one of the highest positive weight cluster-features, consisted of words such as *baked, yummy, chicken* etc. Similarly, several humor-themed Twitter handles such as *@thecollegelife, @collegehumor_, @thecomedyhumor* etc. were highly weighted in the negative class, i.e. for ages below 21. Thus, our method is able to detect the varying conversational topics across ages.

6.8 Conclusion

In this work we have developed a method using distributed representation of words, for effective detection of age of Twitter users from their generic tweets. The method uses Word2Vec to obtain word embeddings that capture contextual similarity between words, with k-means to cluster these embeddings. The selection of words to cluster is obtained by computing odds ratios of the words in question. This enables us to successfully classify users into < 21 and $>= 21$ years of age. In addition, our approach uncovers the most age-revealing topics of conversation that users of different age groups engage in.

Table 6.2: The most highly weighted cluster features obtained using the feature weight vector β (6.3). The positive and negatively weighted cluster-features are represented here. The clusters correspond to different topics of conversation.

Feature Weight	**Topic**	**Words in the cluster**
Highest Positive Cluster-Features	**Music festivals and award shows**	*#coachella2015, #kcas, #disneydescendants, #ytff, #straya, #fooofamily, #theasianawards, #asifbymagic, #getfree, #coachella, #handwrittenbuyouts, #1989tourtokyo, #14days, #tourlife, @benw, #fangirl, #pumped, #xfactor, #idol, #mtv, #americanidol, #nashville*
	Food	*yummy, baked, delicious, shrimp, craving, steak, broccoli, soup, macaroni, chicken*
	Family-related occasions	*mothersday, fathersday, freebies, winwednesday, valentinesday, fridayfreebie, fridayfeeling, kids, winit, valentines, backtoschool, easter, bbq*
Highest Negative Cluster-Features	**School-related**	*biology, gcse, exam, math, assignment, chemistry, accounting*
	Young celebrities (Comedians, Youtubers, Music artists, etc.)	*@sammywilk, @nashgrier, @jakefoushee, @bryanteslava, @twankuyper, @skatemaloley, @jackandjackreal, #asknacks*
	Popular humor-themed Twitter handles	*@factsaboutboys, @comedyortruth, @omgrelatabie, @adorablewords, @lmao, @justagirithing, @speakcomedy, @femalestruggies, @awesomityfun, @thecoliegelife, @teenagernotes, @_collegehumor_, @thatbucketlist, @fillwerrell, @comedypedia, @comedytruth, @thecomedyhumor, @comedyandtruth, @comedyposts*

Chapter 7

Conclusion

This thesis discussed the importance of analyzing online text data to gain a deeper understanding of user opinions and behavior using data mining and machine learning techniques. The challenges associated with extracting meaningful information from online text include abundance of noise, ambiguity, sparsity, and the temporal nature of conversations. We described, in detail the research problems undertaken as part of the author's Ph.D. program, and the methods proposed to address them. First, we proposed a method for opinion mining of Twitter users on a given topic, from their tweets. Keeping in mind the amount of noise and ambiguity present in the text, we were able to develop a supervised machine-learning method that captures user opinions on two different topics successfully, with the use of hashtags and n-grams as features. Second, we addressed the problem of detecting opinions on a conversation which is temporal in nature, i.e. in which the issues pertaining to the topic discussed change over time. Absence of ground truth labels at every timestep makes a typical supervised learning based approach infeasible. We developed a method that has its roots in sociological literature, based on the key observation that temporal evolution of user opinions is a slow process. Our method is able to capture evolving opinions on two different topics, with high accuracies.

The third research problem was that of Aspect-Based Sentiment Analysis on user-generated

reviews of products and services. We aimed to understand, the *what* and *why* behind user likes and dislikes of a product. Using contextual similarity between words captured using distributed word representations, we were able to develop a pipeline that can discover the aspects of products and services that users address in their reviews, and the associated sentiment. Our method has been tested on two different datasets - one on Amazon digital camera reviews, and the other on Yelp restaurant reviews, and performs well on both. The final research problem was that of detecting user attributes from their online posts. In particular, we detected the age of Twitter users from their publicly available tweets, not specific to any one topic. We harnessed contextual similarity amongst words in order to obtain the various topics of conversation that users of different age groups engage in, and used that information to detect user age. We obtained high accuracies for this task, and simultaneously uncovered the most age-discriminating topics of conversation amongst the different age groups.

Appendix A

Additional Experiments for Aspect-Based Sentiment Analysis on Digital Camera Reviews From Amazon

In this Appendix, we demonstrate the generalizability of our Aspect-Based Sentiment Analysis method (Chapter 5) by applying it on a different review dataset. The source of the dataset is Amazon, and we used digital camera reviews for our experiments. The methodology is the same as elaborated in Chapter 5, and we illustrate the results obtained in this Chapter.

A.1 Dataset

The dataset we used is a subset of the Amazon product review dataset [182, 183]. The dataset contains product reviews and metadata from Amazon, including 143.7 million reviews spanning May 1996 - July 2014. The metadata consists of product descriptions, category information, price, brand. etc. We extracted reviews that pertain to digital cameras by exploring the category information in the metadata. Our dataset is comprised of 204,240 reviews. For the purpose of this work, we extracted the text of the review, the summary of the review and the numerical rating (1.0-5.0) of the product. Table A.1 shows a few examples of the reviews.

A.2 Methodology

As mentioned earlier, the method used is similar to that elaborated in Chapter 5. The seed Aspect and Descriptor words selected are domain-specific, hence different from those obtained from the Yelp dataset used earlier. We have 17 Aspect seed words and 53 Descriptor seed words in this case. Table A.2 presents instances of seed words from this dataset and the contextually closest words obtained for these seed words using Word2Vec. Table A.3 illustrates the next step in the pipeline, i.e. the sub-groups obtained from the seed words, that are then used to construct meta-features.

Table A.1: Example of product reviews from the Amazon dataset. The words in bold indicate noise in the text. Noise includes mis-spellings, case insensitivity, misplaced punctuation marks etc.

Summary	**Review excerpt**	**Rating**
Don't buy	I owned this camera for less than 5 hours. I bought it...and decided it was one of the worst purchases that i've ever made and returned it the same night!..... the **battry** consumption is insane.	1.0
For a **NON-digital** Great Grandma...	... after two previous attempts at buying my mother a decent digital camera with ease of operation, battery charge and a basic point and click with flash - this one **FINALLY** hit the mark! **YEAH**! She's 68 and has a bit of difficulty with dexterity and understanding how to make this "thing" work smile. **Fortunatly**, she is able to take the camera to the local drug store if necessary and they will pull out her memory card and let her go through her own picture taking,...	5.0
How do you spell **E-A-S-Y**?	I **recieved** this camera a couple of days ago and I am very **imprissed** at the ease of it and the **sftware**. The quality of the pics are really stunning...I am looking like a professional photographer **alreaddy**, and this is the first time I have picked up a "real" camera since high school photography class...25 years ago...sigh; I have really checked out the prices online and off, and this is a great buy! I don't know how you do it Amazon. Keep bringing us the BARGAINS! Well, I am off to buy a compact flash memory card...ta ta!	3.0
Good value, **usibility** problems, poor Mac interface	The camera produces excellent pictures, but has some **usibility** problems. For a two-megapixel camera, the price is excellent. The **usibility** problems vary in seriousness. The power switch has only one resting **positon** and hence doesn't indicate the power-up status of the camera....	3.0

Table A.2: Instances of seed words and some of their contextually closest words, computed using cosine similarity after Word2Vec training. Misspellings are in bold.

Type of Word	**Seed Word**	**Contextually Closest Words**
Aspect	*picture*	photo, pic, image, pictures, picture's, photos, photographs, **pictue**, pics, images, **picure**, photo's
	aperture	**aperature**, f-stop, **apeture**, fstop, **apature**, priority, 1/60, f-stop, **apperture**
	price	prices, pricing, cost, bargain, value, $399, price-point, pricepoint
Descriptor	*sharp*	crisp, clear, colorful, vibrant, well-exposed, crystal, stunning, lifelike, clean, well-focused
	good	remarkable, impressive, fabulous, suitable, fantastic, excellent, **excellant**
	dying	draining, drained, exhausted, depleted, discharging, drainage, discharge, fail, eating

Table A.3: A few Aspect and Descriptor sub-groups that were used to build Meta-features.

Word Type	**Seed Word**	**Instances of Words in the Subgroup**
Aspect	*navigation*	controls, buttons, levers, knobs, menu, menues, submenus, command, dials, scroll
	memory	tm, gb, 133x,mmc, udma, multimedia, mem, sandisk, transcend, compactflash, memeory, card, 90mb, mg, 16gb, 128m, sim, gigabyte, sm, 8gb, 2g,sdcard, 512, ram, sdhc, sdhd, microsdhc, 2gig, 256mb, scandisk, 128mb, 128, flashcard, 1gig, 4gb, cards, 16mb, microsd, 30mb
	durability	sturdiness, longevity, fragility, workmanship, strengths, ruggedness, merits, usefulness, lightness, effectiveness, construction, robustness, reliability
Descriptor	*saturated*	warm, lifeless, blurring, unfocused, unnatural, washed, gradation, pixelated, fuzzy, distorted, oversaturated
	compact	petite, pocketable, slender, tiny, subcompact
	friendly	friendliness, customization, accessable, configurable, streamlined, simplified, effortless, comprehensive, convenient, efficient, uncomplicated

Table A.4: Classifier Metrics using l_2-regularized Logistic Regression

Overall Accuracy (%)	**Precision**	**Recall**	**Specificity**	**AUC**
72.98	0.952	0.723	0.772	0.75

A.3 Classification Results

The usual classification metrics [151] are reported in Table A.4. The metrics reported are accuracy, precision, recall, specificity and AUC. The best results are obtained with the regularization parameter $\lambda = 100.0$. As may be observed, the method exhibits good performance across the classifier metrics, which demonstrates the predictive power of the meta-features.

Using the sign and magnitude of each component of the feature weight vector β (5.5) learned by the classifier, we obtained an explicit sentiment weight for each meta-feature. Positive feature weights indicate that the corresponding meta-feature carries a positive sentiment, while negative feature weights indicate a negative sentiment. In Table A.5, we demonstrate a few meta-features that have the highest positive and highest negative weights, and were deemed the most discriminative by the classifier. This helps us in making the distinction between meta-features that are widely used and those that have significant sentiment-carrying capacity. For instance, *(picture, quality)* is a popularly used meta-feature, however, it had a very low negative score in β (5.5). Thus the classifier helps in identification of meaningful, sentiment-carrying meta-features, thereby enabling us to understand consumer sentiment at a more granular level.

A.4 Coverage of meta-features

As mentioned in Section 5.6.3, one of the most significant achievements of our method is that the meta-features allowed us to cover a larger number of reviews than if we were to use individual seed words as features. The meta-features were constructed using semantically similar words. Thus, their occurrence spanned a large number of reviews. To illustrate this, we computed the occurrence of tokens that are built *only using the seed words*, i.e., an Aspect seed word and a Descriptor seed word in the neighborhood of the aspect seed word. We picked out the most frequently occurring such tokens, and plot the coverage they yield against the coverage obtained by using the actual meta-features for the corresponding seed words in Figure A.1. For instance, for the seed word token *(picture, good)*, the actual meta-feature accounts for all occurrences of tokens such as *(image, excellent)*, *(pic, astounding)*, *(picture, great)*, etc. Figure A.1 clearly illustrates the larger coverage obtained using the meta-features.

Table A.5: The 10 most positive and negative meta features uncovered from the feature weights during classifier training.

Sentiment of Meta Features	**Meta Features**
Positive	(price, excellent), (picture, easy), (price, best), (settings, simple), (lens, enjoy), (focus, speedy), (case, roomy), (size, perfect), (flash, good), (battery, excellent)
Negative	(picture, lousy), (picture, pixelated), (battery, questionable), (focus, inconsistent), (settings, bad), (flash, poor), (video, bad), (size, bad), (aperture, poor), (battery, depleted)

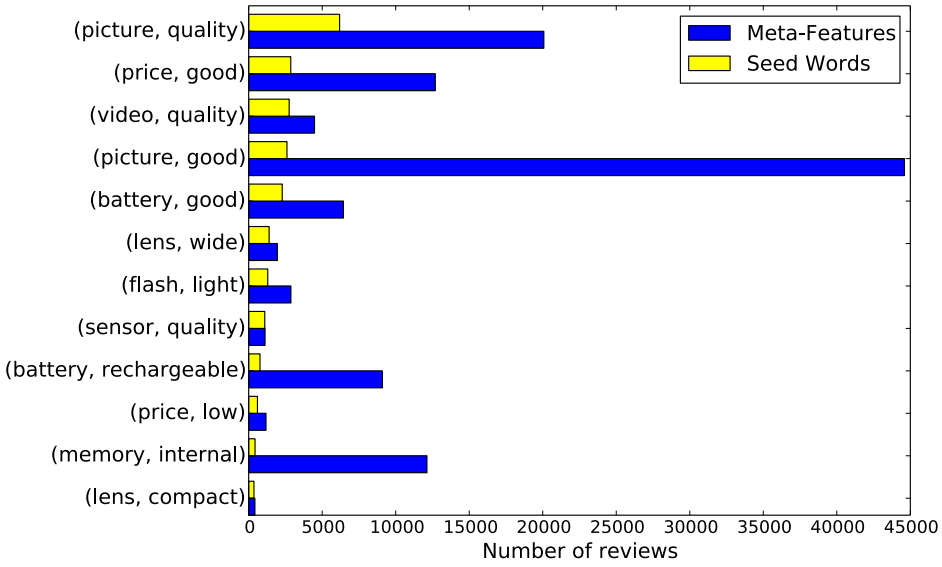


Figure A.1: Occurrence counts of Aspect, Descriptor seed word pairs, as compared to the occurrence counts of their corresponding meta-features. Larger coverage is obtained using meta-features.

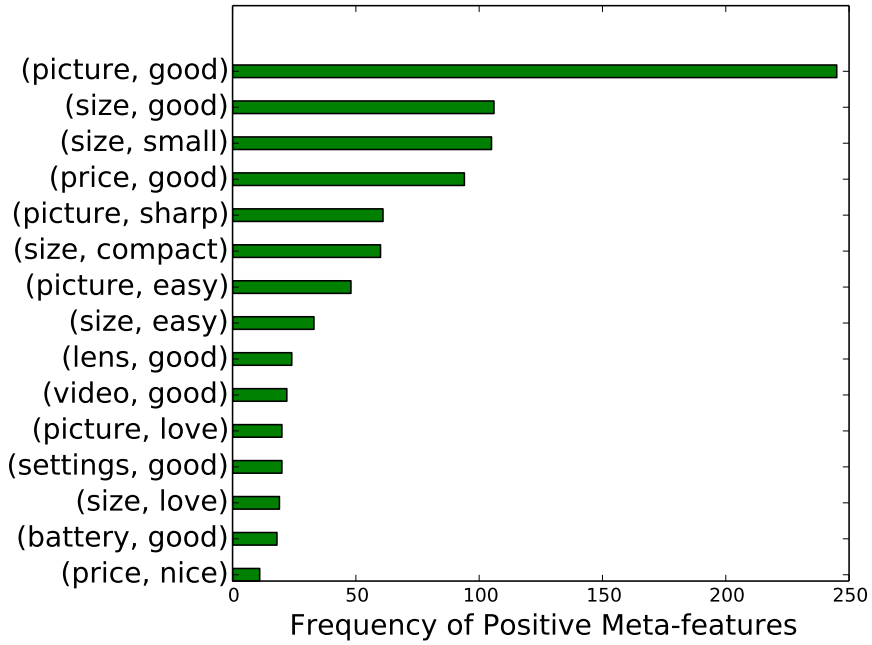


Figure A.2: Frequency of positive meta-features in a point-and-shoot camera - Canon PowerShot A2300 16.0 MP Digital Camera with 5x Optical Zoom (Silver)

A.5 Product-level Summarization

We perform additional summarization tasks on this dataset, at the product-level, and present them in this section. We look at individual products from the review data and present summarization results for each. For each of the products, we generate plots showing the occurrences of discriminative meta-features, as obtained using the classifier. For every product, we obtain the frequency of occurrence of any of the top 50 most discriminative meta-features of each sentiment, and plot the 15 most frequent meta-features. We selected products from different brands (Canon, Nikon) and also picked different products as well (Point and Shoot Camera and DSLR) in order to capture as large a variety as possible.

1. **Canon PowerShot A2300 16.0 MP Digital Camera with 5x Optical Zoom (Silver):** This is a point-and-shoot camera. Figures A.2 and A.3 demonstrate the frequency of occurrence of the most discriminative positive and negative meta-features, respectively, for this product. As is evident from Figure A.2, most users like the pictures taken by the camera [*(picture, good), (picture, sharp)*], and the price [*(price, good), (price, nice)*], followed by the compact size of the camera [*(size, good), (size, small)*]. The product had far more positive sentiment meta-features than negative, and the main cons of the product seem to be that some users found the pictures to not be well-lit [*(picture, dark)*], and the battery to not last very long [*(battery, drained)*].

   We present here some actual instances of reviews pertaining to this product, with the

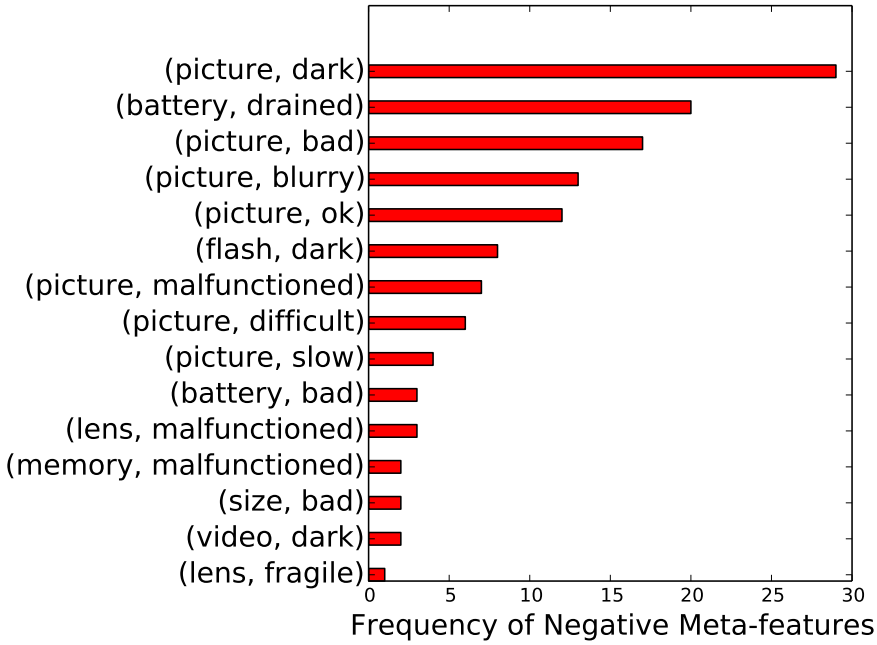


Figure A.3: Frequency of negative meta-features in a point-and-shoot camera - Canon PowerShot A2300 16.0 MP Digital Camera with 5x Optical Zoom (Silver)

words in bold indicating the meta-features.

"Recommended! I just love it. It's a compact camera with all the basic (non professional) functionalities. ***Great*** *for the* ***price****. Amazing camera."*
"Compact, but ***images*** *are* ***blurry****. This is a nice compact camera and the rechargeable batteries are nice to have, but the* ***images*** *are* ***terrible*** *if there is even the slightest movement from one of the subjects."*

2. **Nikon D3100 DSLR Camera with 18-55mm f/3.5-5.6 AF-S Nikkor Zoom Lens (OLD MODEL):** This Nikon DSLR is one of the two DSLRs we consider. The most discriminative positive and negative meta-features for this product are illustrated in Figures A.4 and A.5 respectively. Users were pleased with the pictures taken with the camera [*(picture, good), (picture, sharp)*] and the cost of the camera [*(price, good), (price, cheapest)*]. The cons, fewer in number, seem to be the time taken by the camera to focus [*(focus, slow), (focus, bad)*], the battery drainage [*(battery, drainage)*] and a malfunctioning memory [*(memory, malfunctioned)*].

   Further, we present a few examples of actual reviews of this product from our dataset with the meta-features in bold.
   "...it is just wonderful! It works like a champ and takes the ***clearest pictures****. If only I were a better photographer, but that's not the camera's fault... love it!!!" "....video mode is problematic- despite having 1080p, auto* ***focusing*** *is* ***noisy*** *and* ***slow****, and there is no external mic jack. a serious omission....."*

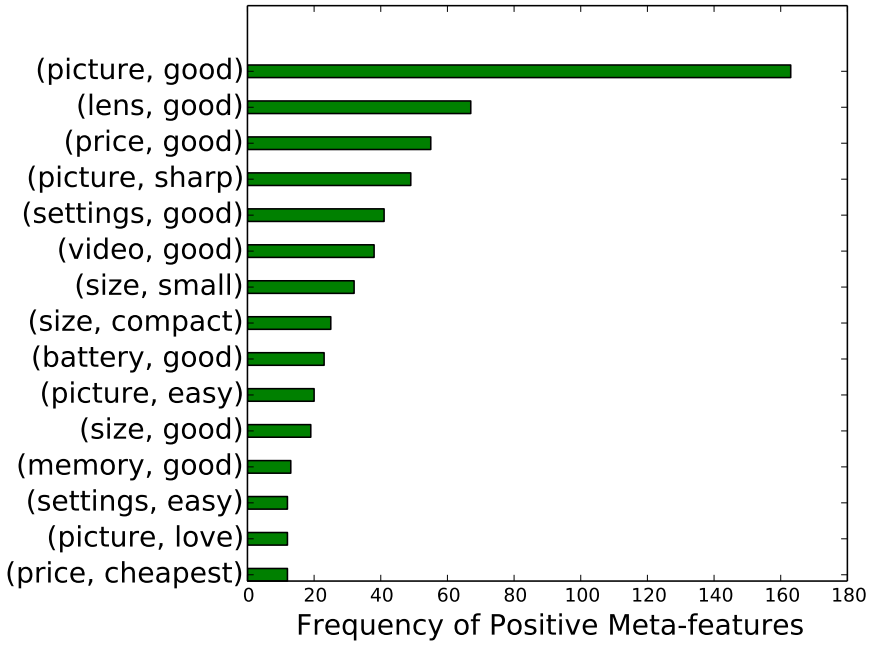


Figure A.4: Frequency of positive meta-features - Nikon D3100 DSLR Camera with 18-55mm f/3.5-5.6 AF-S Nikkor Zoom Lens (OLD MODEL)

3. **Canon EOS Rebel T3i 18 MP CMOS Digital SLR Camera with EF-S 18-55mm f/3.5-5.6 IS Lens (DISCONTINUED):** This is the second DSLR we chose for summarization. For this camera, we have a larger number of reviews as compared to the previously mentioned Nikon DSLR. Figures A.6 and A.7 show the frequency of occurrence of the positive and negative meta features respectively for this camera. Some of the features of the camera that users seem to enjoy were the ease of the settings [*(settings,good)*], the lens quality [*(lens, good), (lens,sharp)*] and the quality of the pictures [*(picture, good), (picture, love)*]. Amongst the aspects of the camera that users disliked are the quality of the videos it took [*(video, dark), (video, bad)*], and the difficulty in focusing [*(focus, difficult)*].

   We present a few review examples with words in bold indicating the meta-features:
   *"Love it! This is one of the best cameras out there at a decent price for recording video. All of your **settings** are super **easy** to adjust and there are lots of good, free instructional videos available on the web."*
   *"Amazing dslr! I got this camera as a christmas present, and i have never been happier with a camera. The Canon T3i is an amazing dslr for beginners or photography enthusiasts, or anyone! the **lens** is a **great** starter **lens**...."*
   *"...this is a great camera for still photography, but for **video** the **limitations** on clip length, sound, ergonomics, and rolling shutter jello just don't cut it for me...."*

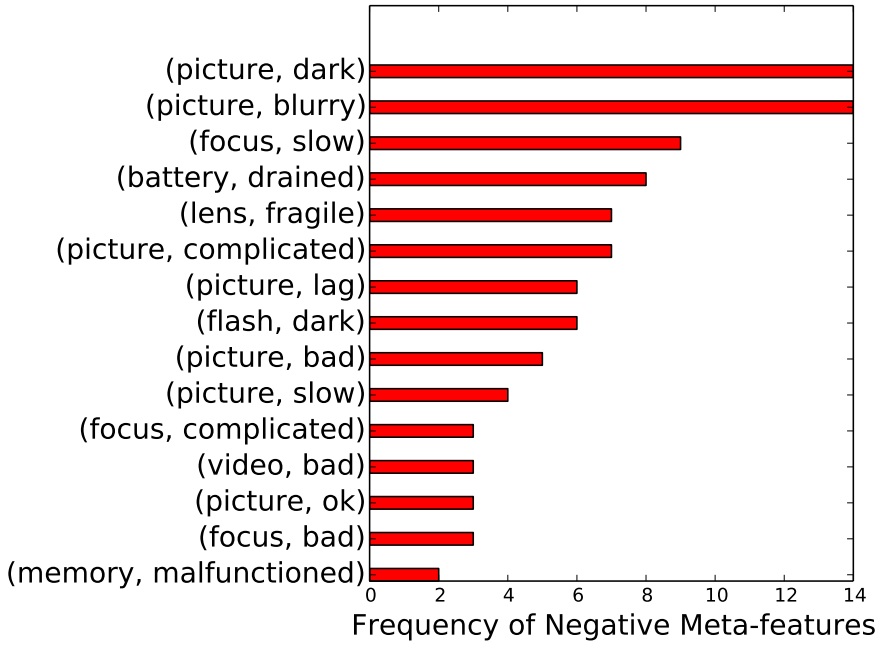


Figure A.5: Frequency of negative meta-features - Nikon D3100 DSLR Camera with 18-55mm f/3.5-5.6 AF-S Nikkor Zoom Lens (OLD MODEL)

A.6 Aspect-level Comparison of Individual Products

An interesting comparison between the products is presented in Figure A.8. We gather the occurrences of all meta-features pertaining to each of the 17 aspects we have, for two of the products - the Canon Point-And-Shoot Camera and the Nikon DSLR Camera. We obtain the sentiment of the meta-features from the feature weight vector β (5.5) and plot the aggregated sentiment per aspect. The plot enables us to observe the differences in the aspects that are most widely discussed in each case, and the corresponding sentiments.

It is interesting to observe that for the Point-And-Shoot Camera, the *size* of the product is widely discussed whereas it is not a very popularly discussed aspect for the DSLR. This is expected since the handiness and compact size of point-and-shoot cameras is one of the main reasons why people invest in them. The *picture* is an important aspect for both types of cameras because that is the primary function of a camera, and is the main parameter users judge a camera by. However the point-and-shoot reviewers were more likely to be satisfied with the picture. They were also more likely to be satisfied with the *price* paid. Otherwise, many camera features *(flash, lens, sensor, video, settings)* have a higher representation in the DSLR review set than the point-and-shoot, which may indicate a higher level of domain knowledge and expectations from DSLR purchasers.

Thus, the meta-features provide us with a powerful way of capturing the most important aspects for a product, and clearly visualize the different aspects that users evaluate different products by.

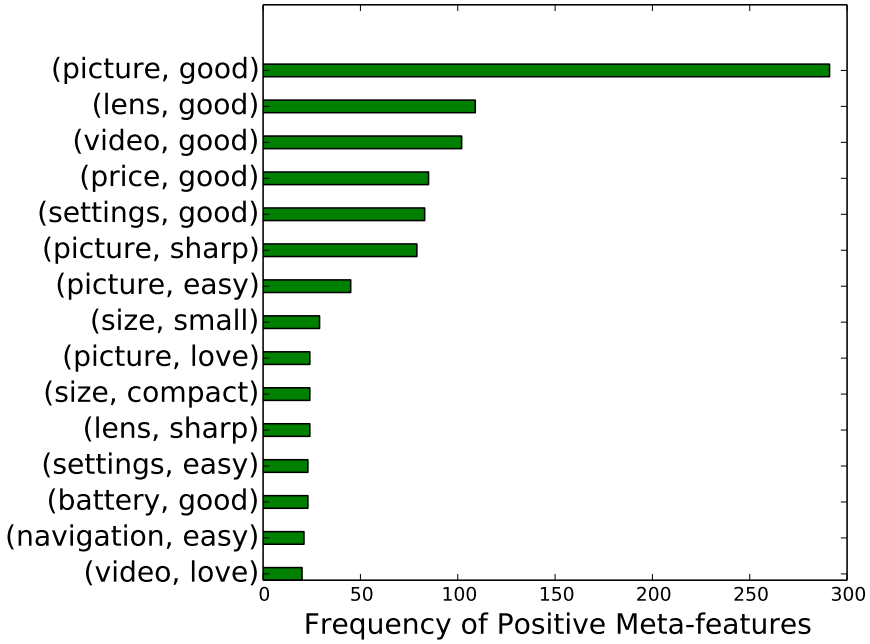


Figure A.6: Frequency of positive meta-features - Canon EOS Rebel T3i 18 MP CMOS Digital SLR Camera with EF-S 18-55mm f/3.5-5.6 IS Lens (DISCONTINUED)

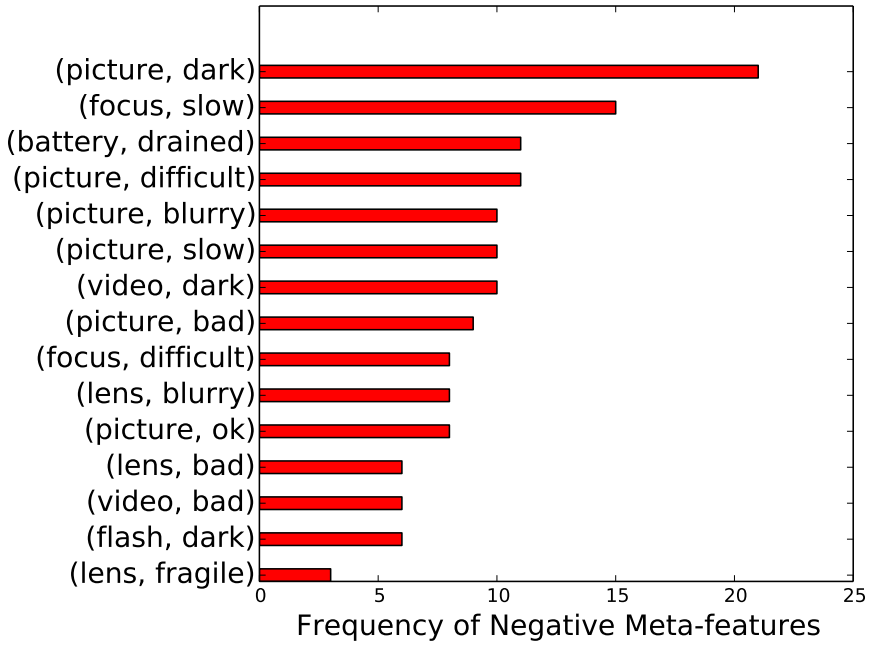


Figure A.7: Frequency of negative meta-features - Canon EOS Rebel T3i 18 MP CMOS Digital SLR Camera with EF-S 18-55mm f/3.5-5.6 IS Lens (DISCONTINUED)

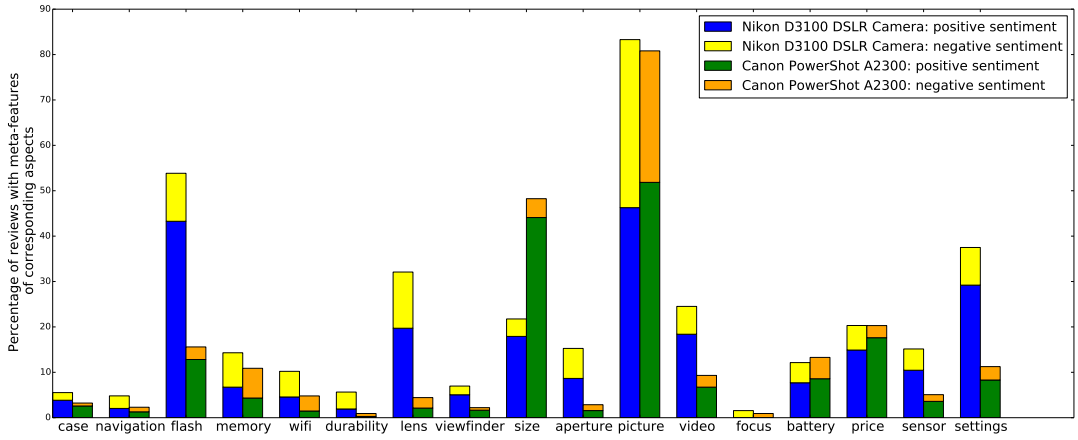


Figure A.8: Comparing Aspect-level sentiment of two cameras - a DSLR and a point-and-shoot.

Bibliography

[1] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*, vol. 8. Cambridge university press, 1994.

[2] M. Zhang, *Social network analysis: history, concepts, and research*. Springer, 2010.

[3] I. T. Union, "Key ict indicators for developed and developing countries and the world (totals and penetration rates), international telecommunication union (itu)," 2015.

[4] "Amazon." www.amazon.com.

[5] "Yelp." https://www.yelp.com/.

[6] "Trip advisor." www.tripadvisor.com.

[7] Y. Marzouki and O. Oullier, "Revolutionizing revolutions: Virtual collective consciousness and the arab spring." http://www.huffingtonpost.com/yousri-marzouki/revolutionizing-revolutio_b_1679181.html, 2012.

[8] P. Rutledge, "How obama won the social media battle in the 2012 presidential campaign." http://mprcenter.org/blog/2013/01/25/how-obama-won-the-social-media-battle-in-the-2012-presidential-campaign, 2013.

[9] A. Galland, "Moveon.org." http://front.moveon.org/thank-you-for-an-awesome-2013/#.Uty0RnmttFQ, 2013.

[10] "Avaaz:the world in action." http://www.avaaz.org/en/highlights.php.

[11] "Twitter." https://twitter.com.

[12] N. A. Christakis and J. H. Fowler, "The collective dynamics of smoking in a large social network," in *New England Journal of Medicine*, vol. 358, pp. 2249–2258, Mass Medical Soc, 2008.

[13] S. Gunelius, "The data explosion in 2014 minute by minute - infographic." http://aci.info/2014/07/12/the-data-explosion-in-2014-minute-by-minute-infographic/, July 2014.

[14] E. Berger, "Dynamic monopolies of constant size," in *Journal of Combinatorial Theory, Series B*, vol. 83, pp. 191–200, Elsevier, 2001.

[15] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, pp. 137–146, ACM, 2003.

[16] J. Goldenberg, B. Libai, and E. Muller, "Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata," in *Academy of Marketing Science Review*, vol. 2001, p. 1, Academy of Marketing Science Review, 2001.

[17] "Leading social networks worldwide as of april 2016, ranked by number of active users (in millions)." `http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/`.

[18] P. A. Dow, L. A. Adamic, and A. Friggeri, "The anatomy of large facebook cascades.," in *International AAAI Conference on Web and Social Media*, 2013.

[19] S. Kumar, "Analyzing the facebook workload," in *2012 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 111–112, IEEE, 2012.

[20] E. Sun, I. Rosenn, C. Marlow, and T. M. Lento, "Gesundheit! modeling contagion through facebook news feed.," in *International AAAI Conference on Web and Social Media*, 2009.

[21] R. Rogers, "Debanalizing twitter: The transformation of an object of study," in *Proceedings of the 5th Annual ACM Web Science Conference*, pp. 356–365, ACM, 2013.

[22] "Twitter developer site." `https://dev.twitter.com`.

[23] "Sina weibo." `http://weibo.com`.

[24] J. Ong, "China's sina weibo grew 73% in 2012, passing 500 million registered accounts." `http://thenextweb.com/asia/2013/02/21/chinas-sina-weibo-grew-73-in-2012-passing-500-million-registered-accounts/#gref`, 2013.

[25] L. Yu, S. Asur, and B. A. Huberman, "What trends in chinese social media," *arXiv preprint arXiv:1107.3522*, 2011.

[26] Q. Gao, F. Abel, G.-J. Houben, and Y. Yu, "A comparative study of users' microblogging behavior on sina weibo and twitter," in *User Modeling, Adaptation, and Personalization*, pp. 88–101, Springer, 2012.

[27] Z. Guo, Z. Li, and H. Tu, "Sina microblog: an information-driven online social network," in *2011 International Conference on Cyberworlds (CW)*, pp. 160–167, IEEE, 2011.

[28] Y. Qu, C. Huang, P. Zhang, and J. Zhang, "Microblogging after a major disaster in china: a case study of the 2010 yushu earthquake," in *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, pp. 25–34, ACM, 2011.

[29] R. Lu and Q. Yang, "Trend analysis of news topics on twitter," in *International Journal of Machine Learning and Computing*, vol. 2, p. 327, IACSIT Press, 2012.

[30] "Yelp dataset challenge." `https://www.yelp.com/dataset_challenge`.

[31] "Stanford snap infolab." `https://snap.stanford.edu/data/web-Amazon.html`.

[32] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet Measurement*, pp. 29–42, ACM, 2007.

[33] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *Proceedings of the 18th International Conference on World Wide Web*, pp. 721–730, ACM, 2009.

[34] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Link Mining: Models, Algorithms, and Applications*, pp. 337–357, Springer, 2010.

[35] K. Lerman and T. Hogg, "Using a model of social dynamics to predict popularity of news," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 621–630, ACM, 2010.

[36] F. Wu and B. A. Huberman, "Novelty and collective attention," in *Proceedings of the National Academy of Sciences*, vol. 104, pp. 17599–17601, National Acad Sciences, 2007.

[37] J. Diesner, T. L. Frantz, and K. M. Carley, "Communication networks from the enron email corpus "it's always about the people. enron is no different"," in *Computational & Mathematical Organization Theory*, vol. 11, pp. 201–228, Springer, 2005.

[38] L. A. Adamic and E. Adar, "Friends and neighbors on the web," in *Social Networks*, vol. 25, pp. 211–230, Elsevier, 2003.

[39] L. Adamic and E. Adar, "How to search a social network," in *Social Networks*, vol. 27, pp. 187–203, Elsevier, 2005.

[40] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proceedings of the 13th International Conference on World Wide Web*, pp. 491–501, ACM, 2004.

[41] D. Lopresti, S. Roy, K. Schulz, and L. V. Subramaniam, "Special issue on noisy text analytics," in *International Journal on Document Analysis and Recognition*, vol. 12, pp. 139–140, Springer, 2009.

[42] L. V. Subramaniam, S. Roy, T. A. Faruquie, and S. Negi, "A survey of types of text noise and techniques to handle noisy text," in *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, pp. 115–122, ACM, 2009.

[43] K. Bhattacharjee and L. Petzold, "Probabilistic user-level opinion detection on online social networks," in *Social Informatics*, pp. 309–325, Springer, 2014.

[44] K. Bhattacharjee and L. Petzold, "Detecting opinions in a temporally evolving conversation on twitter," in *Social Informatics*, pp. 82–97, Springer, 2015.

[45] "10th ACM International Conference on Web Search and Data Mining." `http://www.wsdm-conference.org/2017/`.

[46] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: modeling facets and opinions in weblogs," in *Proceedings of the 16th international conference on World Wide Web*, pp. 171–180, ACM, 2007.

[47] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a maxent-lda hybrid," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 56–65, Association for Computational Linguistics, 2010.

[48] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, ACM, 2004.

[49] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Natural Language Processing and Text Mining*, pp. 9–28, Springer, 2007.

[50] M. D. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Menczer, "Predicting the political alignment of twitter users," in *Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on Social Computing (SocialCom)*, pp. 192–199, IEEE, 2011.

[51] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," in *CS224N Project Report, Stanford*, pp. 1–12, 2009(a).

[52] B. Pang and L. Lee, "Opinion mining and sentiment analysis," in *Foundations and Trends in Information Retrieval*, vol. 2, pp. 1–135, Now Publishers Inc., 2008.

[53] B. Liu and L. Zhang, *A survey of opinion mining and sentiment analysis*. Springer, 2012.

[54] J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, and R. J. Booth, "The development and psychometric properties of liwc2007," in *Austin, TX, LIWC. Net*, 2007.

[55] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," in *Mahway: Lawrence Erlbaum Associates*, p. 71, 2001.

[56] MPQA, "MPQA." `http://mpqa.cs.pitt.edu/lexicons/`, 2005.

[57] T. Wilson, *Fine-grained subjectivity analysis*. PhD thesis, Doctoral Dissertation, University of Pittsburgh, 2008.

[58] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," in *Language Resources and Evaluation*, vol. 39, pp. 165–210, Springer, 2005.

[59] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," in *Journal of the American Society for Information Science and Technology*, vol. 61, pp. 2544–2558, Wiley Online Library, 2010.

[60] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining.," in *LREC*, vol. 10, pp. 2200–2204, 2010.

[61] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of LREC*, vol. 6, pp. 417–422, 2006.

[62] K. Denecke, "Using sentiwordnet for multilingual sentiment analysis," in *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*, pp. 507–512, IEEE, 2008.

[63] A. Garas, D. Garcia, M. Skowron, and F. Schweitzer, "Emotional persistence in online chatting communities," in *Scientific Reports*, vol. 2, Nature Publishing Group, 2012.

[64] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, "A large-scale sentiment analysis for yahoo! answers," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pp. 633–642, ACM, 2012.

[65] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in twitter events," in *Journal of the American Society for Information Science and Technology*, vol. 62, pp. 406–418, Wiley Online Library, 2011.

[66] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, vol. 10, pp. 79–86, Association for Computational Linguistics, 2002.

[67] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th International Conference on World Wide Web*, pp. 519–528, ACM, 2003.

[68] C. Whitelaw, N. Garg, and S. Argamon, "Using appraisal groups for sentiment analysis," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pp. 625–631, ACM, 2005.

[69] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources.," in *Empirical Methods on Natural Language Processing*, vol. 4, pp. 412–418, 2004.

[70] T. Kudo and Y. Matsumoto, "A boosting algorithm for classification of semi-structured text.," in *Empirical Methods on Natural Language Processing*, vol. 4, pp. 301–308, 2004.

[71] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 375–384, ACM, 2009.

[72] S. Brody and N. Elhadad, "An unsupervised aspect-sentiment model for online reviews," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 804–812, Association for Computational Linguistics, 2010.

[73] C. Scaffidi, K. Bierhoff, E. Chang, M. Felker, H. Ng, and C. Jin, "Red opal: product-feature scoring from reviews," in *Proceedings of the 8th ACM Conference on Electronic Commerce*, pp. 182–191, ACM, 2007.

[74] M. Cheong and V. Lee, "Integrating web-based intelligence retrieval and decision-making from the twitter trends knowledge base," in *Proceedings of the 2nd ACM Workshop on Social Web Search and Mining*, pp. 1–8, ACM, 2009.

[75] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 1155–1158, ACM, 2010.

[76] J. Benhardus and J. Kalita, "Streaming trend detection in twitter," in *International Journal of Web Based Communities*, vol. 9, pp. 122–139, Inderscience Publishers Ltd, 2013.

[77] A. Culotta, "Towards detecting influenza epidemics by analyzing twitter messages," in *Proceedings of the First Workshop on Social Media Analytics*, pp. 115–122, ACM, 2010.

[78] M. J. Paul and M. Dredze, "A model for mining public health topics from twitter," in *Health*, vol. 11, pp. 16–6, 2012.

[79] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," in *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, JMLR. org, 2003.

[80] M. J. Paul and M. Dredze, "You are what you tweet: Analyzing twitter for public health.," in *International AAAI Conference on Web and Social Media*, vol. 20, pp. 265–272, 2011.

[81] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 851–860, ACM, 2010.

[82] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[83] J. Teevan, S. T. Dumais, and E. Horvitz, "Personalizing search via automated analysis of interests and activities," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 449–456, ACM, 2005.

[84] S. Kanoje, S. Girase, and D. Mukhopadhyay, "User profiling trends, techniques and applications," in *arXiv preprint arXiv:1503.07474*, 2015.

[85] D.-P. Nguyen, R. Gravel, R. Trieschnigg, and T. Meder, "" how old do you think i am?" a study of language and age in twitter," in *AAAI Press*, 2013.

[86] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *Proceedings of the 16th International Conference on World Wide Web*, pp. 271–280, ACM, 2007.

[87] H.-N. Kim, I. Ha, K.-S. Lee, G.-S. Jo, and A. El-Saddik, "Collaborative user modeling for enhanced content filtering in recommender systems," in *Decision Support Systems*, vol. 51, pp. 772–781, Elsevier, 2011.

[88] C. Lu, W. Lam, and Y. Zhang, "Twitter user modeling and tweets recommendation based on wikipedia concept graph," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[89] K. Tao, F. Abel, Q. Gao, and G.-J. Houben, "Tums: Twitter-based user modeling service," in *The Semantic Web: ESWC 2011 Workshops*, pp. 269–283, Springer, 2011.

[90] M. H. DeGroot, "Reaching a consensus," in *Journal of the American Statistical Association*, vol. 69, pp. 118–121, Taylor & Francis, 1974.

[91] N. E. Friedkin and E. C. Johnsen, "Social influence networks and opinion change," in *Advances in Group Processes*, vol. 16, pp. 1–29, 1999.

[92] S. Moscovici, "Social influence and conformity," in *The Handbook of Social Psychology*, vol. 2, Random House, 1985.

[93] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy.," in *International AAAI Conference on Web and Social Media*, vol. 10, p. 30, 2010.

[94] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *Proceedings of the third ACM International Conference on Web Search and Data Mining*, pp. 261–270, ACM, 2010.

[95] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web.," in *Stanford InfoLab*, Stanford InfoLab, 1999.

[96] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pp. 65–74, ACM, 2011.

[97] M. Granovetter, "Threshold models of collective behavior," in *American Journal of Sociology*, pp. 1420–1443, JSTOR, 1978.

[98] T. C. Schelling, *Micromotives and macrobehavior*. WW Norton & Company, 2006.

[99] M. W. Macy, "Chains of cooperation: Threshold effects in collective action," in *American Sociological Review*, pp. 730–747, JSTOR, 1991.

[100] M. W. Macy and R. Willer, "From factors to actors: Computational sociology and agent-based modeling," in *Annual Review of Sociology*, pp. 143–166, JSTOR, 2002.

[101] S. Morris, "Contagion," in *The Review of Economic Studies*, vol. 67, pp. 57–78, Oxford University Press, 2000.

[102] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," in *Marketing Letters*, vol. 12, pp. 211–223, Springer, 2001.

[103] S. Bikhchandani, D. Hirshleifer, and I. Welch, "A theory of fads, fashion, custom, and cultural change as informational cascades," in *Journal of Political Economy*, pp. 992–1026, JSTOR, 1992.

[104] E. M. Rogers, *Diffusion of innovations*. Simon and Schuster, 2010.

[105] V. Mahajan, E. Muller, and F. M. Bass, "Diffusion of new products: Empirical generalizations and managerial uses," in *Marketing Science*, vol. 14, pp. G79–G88, INFORMS, 1995.

[106] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," in *Reviews of Modern Physics*, vol. 87, p. 925, APS, 2015.

[107] M. E. Newman, "Spread of epidemic disease on networks," in *Physical Review E*, vol. 66, p. 016128, APS, 2002.

[108] R. M. Anderson, R. M. May, and B. Anderson, *Infectious diseases of humans: dynamics and control*, vol. 28. Wiley Online Library, 1992.

[109] O. Diekmann, H. Heesterbeek, and T. Britton, *Mathematical tools for understanding infectious disease dynamics*. Princeton University Press, 2012.

[110] M. J. Keeling and P. Rohani, "Estimating spatial coupling in epidemiological systems: a mechanistic approach," in *Ecology Letters*, vol. 5, pp. 20–29, Wiley Online Library, 2002.

[111] W. Goffman, "Mathematical approach to the spread of scientific ideas–the history of mast cell research.," in *Nature*, vol. 212, pp. 449–452, 1966.

[112] W. Goffman and V. Newill, "Generalization of epidemic theory," in *Nature*, vol. 204, pp. 225–228, 1964.

[113] D. J. Daley and D. G. Kendall, "Epidemics and rumours," in *Nature*, vol. 204, Nature Publishing Group, 1964.

[114] P. Clifford and A. Sudbury, "A model for spatial conflict," in *Biometrika*, vol. 60, pp. 581–588, Biometrika Trust, 1973.

[115] R. A. Holley and T. M. Liggett, "Ergodic theorems for weakly interacting infinite systems and the voter model," in *The Annals of Probability*, pp. 643–663, JSTOR, 1975.

[116] K. Sznajd-Weron and J. Sznajd, "Opinion evolution in closed community," in *International Journal of Modern Physics C*, vol. 11, pp. 1157–1165, World Scientific, 2000.

[117] G. Weisbuch, G. Deffuant, F. Amblard, and J.-P. Nadal, "Meet, discuss, and segregate!," in *Complexity*, vol. 7, pp. 55–63, Wiley Online Library, 2002.

[118] R. Hegselmann, U. Krause, *et al.*, "Opinion dynamics and bounded confidence models, analysis, and simulation," in *Journal of Artificial Societies and Social Simulation*, vol. 5, Citeseer, 2002.

[119] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, ACM, 2001.

[120] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of the 2007 ACM Conference on Recommender Systems*, pp. 17–24, ACM, 2007.

[121] Y. Koren, "Collaborative filtering with temporal dynamics," in *Communications of the ACM*, vol. 53, pp. 89–97, ACM, 2010.

[122] R. R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends.," in *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, vol. 1, 2001.

[123] J. Golbeck, *Generating predictive movie recommendations from trust in social networks*. Springer, 2006.

[124] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 931–940, ACM, 2008.

[125] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 397–406, ACM, 2009.

[126] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the Fourth ACM International conference on Web Search and Data Mining*, pp. 287–296, ACM, 2011.

[127] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," in *Social Network Analysis and Mining*, vol. 3, pp. 1113–1133, Springer, 2013.

[128] P. Victor, M. De Cock, and C. Cornelis, *Trust and recommendations*. Springer, 2011.

[129] P. Victor, C. Cornelis, M. De Cock, and A. Teredesai, "A comparative analysis of trust-enhanced recommenders for controversial items.," in *International AAAI Conference on Web and Social Media*, 2009.

[130] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pp. 492–508, Springer, 2004.

[131] J. Tang, H. Gao, X. Hu, and H. Liu, "Exploiting homophily effect for trust prediction," in *Proceedings of the sixth ACM International Conference on Web Search and Data Mining*, pp. 53–62, ACM, 2013.

[132] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM Conference on Recommender Systems*, pp. 135–142, ACM, 2010.

[133] A. Franz, "Automatic ambiguity resolution in natural language processing: an empirical approach," *Springer Science & Business Media*, vol. 1171, 1996.

[134] X.-H. Phan, L.-M. Nguyen, and S. 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*, pp. 91–100, ACM, 2008.

[135] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," in *Machine Learning*, vol. 42, pp. 143–175, Springer, 2001.

[136] A. Y. Ng, "Feature election, l1 vs. l2 regularization, and rotational invariance," in *Proceedings of the Twenty-First International Conference on Machine Learning*, p. 78, ACM, 2004.

[137] "LIWC software." `http://www.liwc.net/index.php`, 2001.

[138] C. Akkaya, J. Wiebe, and R. Mihalcea, "Subjectivity word sense disambiguation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, pp. 190–199, Association for Computational Linguistics, 2009.

[139] E. Gilbert and K. Karahalios, "Predicting tie strength with social media," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, (New York, NY, USA), pp. 211–220, ACM, 2009.

[140] J. E. Bono and R. Ilies, "Charisma, positive emotions and mood contagion," in *The Leadership Quarterly*, vol. 17, pp. 317–334, Elsevier, 2006.

[141] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using twitter hashtags and smileys," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 241–249, Association for Computational Linguistics, 2010.

[142] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 1031–1040, ACM, 2011.

[143] S. Tan, Y. Li, H. Sun, Z. Guan, X. Yan, J. Bu, C. Chen, and X. He, "Interpreting the public sentiment variations on twitter," in *IEEE Transactions on Knowledge and Data Engineering*, IEEE, 2012.

[144] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li, "User-level sentiment analysis incorporating social networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1397–1405, ACM, 2011.

[145] M. Pennacchiotti and A.-M. Popescu, "A machine learning approach to twitter user classification.," in *International AAAI Conference on Web and Social Media*, 2011.

[146] "Twitter rest api." `https://dev.twitter.com/rest/public`.

[147] D. M. Romero, B. Meeder, and J. Kleinberg, "Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, pp. 695–704, ACM, 2011.

[148] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," in *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[149] D. Bespalov, B. Bai, Y. Qi, and A. Shokoufandeh, "Sentiment classification based on supervised latent n-gram analysis," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 375–382, ACM, 2011.

[150] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining.," in *LREC*, 2010.

[151] T. Fawcett, "An introduction to ROC analysis," in *Pattern Recognition Letters*, vol. 27, pp. 861–874, Elsevier, 2006.

[152] A. Go, L. Huang, and R. Bhayani, "Twittersentiment." `http://www.sentiment140.com`, 2009(b).

[153] "Obamacare facts." `http://obamacarefacts.com/obamacare-facts/`.

[154] "Obamacare sees social media surge ahead of deadline." `http://www.nextgov.com/health/2014/03/obamacare-sees-social-media-surge-ahead-deadline/81625/`, March 2014.

[155] "U.S. immigration reform." `https://en.wikipedia.org/wiki/Border_Security,_Economic_Opportunity,_and_Immigration_Modernization_Act_of_2013`.

[156] "Twitter streaming api." `https://dev.twitter.com/streaming/overview`.

[157] S. V. Pendse, I. K. Tetteh, F. H. Semazzi, V. Kumar, and N. F. Samatova, "Toward data-driven, semi-automatic inference of phenomenological physical models: Application to eastern sahel rainfall.," in *SIAM International Conference on Data Mining*, pp. 35–46, 2012.

[158] "p value." `http://en.wikipedia.org/wiki/P-value`.

[159] A. B. Wilcox and G. Hripcsak, "The role of domain knowledge in automating medical text report classification," in *Journal of the American Medical Informatics Association*, vol. 10, pp. 330–338, Elsevier, 2003.

[160] D. Gruhl, M. Nagarajan, J. Pieper, C. Robson, and A. Sheth, *Context and domain knowledge enhanced entity spotting in informal text*. Springer, 2009.

[161] Y. S. Chan and H. T. Ng, "Domain adaptation with active learning for word sense disambiguation," *Annual Meeting-Association for Computational Linguistics*, vol. 45, no. 1, p. 49, 2007.

[162] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *arXiv preprint arXiv:1301.3781*, 2013.

[163] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting similarities among languages for machine translation," in *arXiv preprint arXiv:1309.4168*, 2013.

[164] "7 factors consumers consider when choosing a restaurant." `http://www.restaurant.org/News-Research/News/7-factors-consumers-consider-when-choosing-a-resta`.

[165] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pp. 43–50, ACM, 2006.

[166] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," *Advances in Neural Information Processing Systems*, pp. 288–296, 2009.

[167] H. Nakagawa and T. Mori, "A simple but powerful automatic term extraction method," in *COLING-02 on COMPUTERM 2002: Second International Workshop on Computational Terminology*, vol. 14, pp. 1–7, Association for Computational Linguistics, 2002.

[168] "Distributional semantics: Wikipedia." `https://en.wikipedia.org/wiki/Distributional_semantics`.

[169] J. R. Firth, "A synopsis of linguistic theory, 1930-1955," in *Studies in Linguistic Analysis*, Blackwell, 1957.

[170] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, (Valletta, Malta), pp. 45–50, ELRA, May 2010. `http://is.muni.cz/publication/884893/en`.

[171] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[172] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on*

Human Language Technology-Volume 1, pp. 173–180, Association for Computational Linguistics, 2003.

[173] "Worditout." http://worditout.com/.

[174] M. Duggan, "The demographics of social media users." http://www.pewinternet.org/2015/08/19/the-demographics-of-social-media-users/.

[175] "Children ignore age limits by opening social media accounts." http://www.telegraph.co.uk/news/health/children/12147629/Children-ignore-age-limits-by-opening-social-media-accounts.html.

[176] "Minimum age requirements on various social media platforms as of 2014." http://www.adweek.com/socialtimes/social-media-minimum-age/501920.

[177] D.-P. Nguyen, R. Trieschnigg, A. Doğruöz, R. Gravel, M. Theune, T. Meder, and F. de Jong, "Why gender and age prediction from tweets is hard: Lessons from a crowdsourcing experiment," in *Association for Computational Linguistics*, 2014.

[178] E. Forgey, "Cluster analysis of multivariate data: Efficiency vs. interpretability of classification," in *Biometrics*, vol. 21, pp. 768–769, 1965.

[179] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, Oakland, CA, USA., 1967.

[180] "Odds ratio." https://en.wikipedia.org/wiki/Odds_ratio.

[181] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *International Conference on Machine Learning*, vol. 97, pp. 412–420, 1997.

[182] J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2015.

[183] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, ACM, 2015.